Pós-Graduação em Ciência da Computação

Diogo Philippini Pontual Branco

Agrupamento fuzzy c-medoids semi-supervisionado de dados relacionais representados por múltiplas matrizes de dissimilaridade

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

RECIFE
2017

Diogo Philippini Pontual Branco

Agrupamento fuzzy c-medoids semi-supervisionado de dados relacionais representados por múltiplas matrizes de dissimilaridade

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Francisco de Assis Tenório de Carvalho

RECIFE
2017

Catalogação na fonte
Bibliotecária Elaine Cristina de Freitas, CRB4-1790

B816a Branco, Diogo Philippini Pontual.
Agrupamento fuzzy c-medoids semi-supervisionado de dados relacionais representados por múltiplas matrizes de dissimilaridade / Diogo Philippini Pontual Branco. – 2017.
67 f.: il., fig., tab.

Orientador: Francisco de Assis Tenório de Carvalho.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2017.
Inclui referências e apêndice.

1. Inteligência computacional 2. Dados relacionais I. Carvalho, Francisco de Assis Tenório (orientador) II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2017-200

Diogo Philippini Pontual Branco

Agrupamento fuzzy c-medoids Semi-supervisionado de Dados Relacionais Representados por Múltiplas Matrizes de Dissimilaridade

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 21/07/2017

BANCA EXAMINADORA

_____

Prof. Dr. Sergio Ricardo de Melo Queiroz
Centro de Informática / UFPE

_____

Prof. Dr. Marcelo Rodrigo Portela Ferreira
Departamento de Estatística / UFPB

_____

Prof. Dr. Francisco de Assis Tenório de Carvalho
Centro de Informática / UFPE
(**Orientador**)

Dedico esta dissertação à minha família, amigos e professores que me deram o suporte necessário para chegar aqui.

AGRADECIMENTOS

Agradeço à todos os docentes que contribuíram positivamente para minha vida acadêmica até este ponto, em especial ao professor Francisco de Assis Tenório de Carvalho por ter me orientado neste trabalho e pela sua confiança. Agradeço também à minha família por sempre me apoiar e me compreender. Por fim, agradeço à UFPE e à FACEPE que tornaram o presente trabalho possível.

Strive not to be a success, but rather to be of value.
(Albert Einstein)

RESUMO

Técnicas de agrupamento de dados geralmente operam em objetos que podem estar descritos pelos seus atributos (*feature data*) ou por dados relacionais. Em dados relacionais apenas a informação que representa o grau de relacionamento entre os pares de objetos está disponível. O caso mais comum de dados relacionais é quando se tem uma matriz de dissimilaridade (NxN) entre N objetos e cada célula da matriz tem a informação do grau de relacionamento entre um par de objetos. Esses dados relacionais podem ser (e geralmente são) complexos, tais como objetos multimídia, o que faz com que o relacionamento entre objetos possa ser descrito por múltiplas matrizes de (dis)similaridade. Cada matriz é chamada de visão e dados descritos desta forma são ditos *multi-view*. Há três principais abordagens para administrar dados *multi-view* em análise de agrupamento no estado da arte: abordagem de concatenação (fusão de dados), abordagem distribuída e abordagem centralizada. Na abordagem centralizada, se utiliza as múltiplas visões de forma simultânea para encontrar padrões escondidos nos dados; representa um desafio importante pois requer uma modificação profunda do processo de particionamento. Em compensação, essa abordagem geralmente tem uma qualidade dos resultados superior em relação às outras duas abordagens. Agrupamento de dados é uma tarefa difícil, especialmente quando se trata de dados complexos, relacionais, de alta dimensionalidade e com múltiplas visões. Para facilitar o processo, não é incomum utilizar os rótulos dos objetos, contudo, dados rotulados geralmente são escassos; por isso é comum o uso de supervisão parcial, que necessita apenas o rótulo de alguns objetos de um dado conjunto. Este trabalho introduz o algoritmo SS-MVFCVSMdd (*Semi-Supervised Multi-View Fuzzy Clustering Vector Set-Medoids*), baseado no MVFCVSMdd e com funcionamento parecido com o SS-MVFCSMdd. O SS-MVFCVSMdd é um algoritmo particional do tipo *fuzzy c-medoids vectors* semi-supervisionado de dados relacionais representados por múltiplas matrizes de dissimilaridade. O SS-MVFCVSMdd utiliza restrições par-a-par (*must-link* e *cannot-link*) entre objetos como supervisão parcial e é capaz de inferir representantes e pesos de relevância para cada visão. Experimentos são realizados em vários conjuntos de dados comparando seu desempenho com algoritmos de características similares ao SS-MVFCVSMdd. Os resultados apontam que o SS-MVFCVSMdd teve uma qualidade similar ou superior em relação aos outros algoritmos.

Palavras-chaves: Agrupamento Fuzzy. Visão Múltipla. Dados Relacionais. Semi-supervisão.

ABSTRACT

Data clustering techniques generally work with objects that can be described by either feature or relational data. In relational data only the information pertaining the relationship degree between pairs of objects is available. The most usual case of relational data is when there is a dissimilarity matrix (NxN) between N objects and each cell of said matrix contains the relationship degree between a given pair of objects. These relational data may be (and generally are) complex, such as multimedia objects, which may cause the relationship between those objects to be described by multiple (dis)similarity matrices. Each matrix is called view and data described in that way are said to be multi-view. There are three main approaches to manage multi-view data in cluster analysis in the the state of the art: concatenation, distributed and centralized. In the centralized approach the views are considered simultaneously in order to find hidden patterns in the data. On one hand, this poses a great challenge as it requires a profound change in the clustering process. On the other hand, this approach generally offers results with superior quality in comparison with the other two approaches. Clustering is a hard task, specially when it concerns complex relational high-dimension multi-view data. To facilitate the process it is not unusual to use the object labels, although labeled data are generally scarce. Therefore the use of parcial supervision is common, which requires only some of the objects are labeled in a given dataset. This work introduces the SS-MVFCVSMdd (Semi-Supervised Multi-View Fuzzy Clustering Vector Set-Medoids) algorithm, based on the MVFCVS-Mdd and functions in a similar way as the SS-MVFCSMdd. The SS-MVFCVSMdd is a semi-supervised multi-view fuzzy c-medoids vectors partitional algorithm, which utilizes pairwise constraints (*must-link* and *cannot-link*) between objects as partial supervision and infers prototypes and relevance weights for each view. Experiments performed using several datasets comparing the performance of the proposed algorithm with algorithms that have similar characteristics as the proposed algorithm. The results indicate that the SS-MVFCVSMdd had a similar or superior quality than the other algorithms.

Key-words: Fuzzy Partitioning. Multi-view. Relational Data. Semi-supervised.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Agrupamento de dados é uma tarefa essencial e frequentemente usada em reconhecimento de padrões, mineração de dados, visão computacional e bioinformática. Seu objetivo é organizar um conjunto de objetos em grupos tal que objetos contidos num mesmo grupo possuem alto grau de similaridade, enquanto objetos pertencentes à grupos diferentes possuem alto grau de dissimilaridade (JAIN, 2010).

Técnicas de agrupamento podem ser divididas em métodos hierárquicos e particionais. Métodos hierárquicos proveem uma estrutura hierárquica de grupos, ou seja, uma sequência aninhada de partições dos dados de entrada (objetos) frequentemente representados por um dendrograma. Métodos particionais almejam prover uma única partição dos dados de entrada em um número fixo de grupos, comumente através da otimização de uma função objetivo que mede a heterogeneidade dentro dos grupos. Métodos particionais, por sua vez, podem ser divididos em métodos *hard* e *fuzzy*. No método *hard* o agrupamento é estrito, qualquer objeto pode pertencer a um e apenas um grupo. Por outro lado, no método *fuzzy* o agrupamento é não-estrito, os objetos podem ser designados para todos os grupos com um certo grau de pertinência *fuzzy* (JAIN; MURTY; FLYNN, 1999; XU; WUNUSCH, 2005).

Técnicas de agrupamento de dados geralmente operam em objetos que podem estar descritos pelos seus atributos (*feature data*) ou por dados relacionais. Objetos descritos por um vetor de valores quantitativos ou qualitativos representam *feature data*. Enquanto que em dados relacionais apenas a informação que representa o grau de relacionamento entre os pares de objetos está disponível. O caso mais comum de dados relacionais é quando se tem uma matriz de dissimilaridade ($N \times N$) entre N objetos e cada célula da matriz tem a informação do grau de relacionamento entre um par de objetos.

Métodos particionais de agrupamento geralmente trabalham com apenas uma única matriz de *feature data*. Apesar desses métodos terem sido profundamente estudados e serem muito úteis na pŕatica, há uma demanda crescente de métodos que são capazes de trabalhar com objetos descritos por múltiplas visões (*multi-view*), geralmente envolvendo dados extraídos de fontes diferentes com diferentes conjuntos de medidas e escalas. Por exemplo, em estudos de tumores pode ser necessário levar em conta, simultaneamente, dados genômicos, epigenômicos, transcriptômicos e proteico (SHENL; OLSHEN; LADANYI, 2009).

Há três principais abordagens para administrar dados *multi-view* em análise de agrupamento no estado da arte: abordagem de concatenação (fusão de dados), abordagem distribuída e abordagem centralizada. A primeira consiste na concatenação das visões em uma única visão, seja justapondo o conjunto de características ou combinando, indi-

retamente, as matrizes de proximidade derivadas de cada visão. Já a segunda, também conhecido por agregação de grupo (*cluster ensemble*), agrupa os objetos de cada visão de forma independente e então procura por uma solução que representa um consenso entre o conjunto de grupos; a principal desvantagem desse método é que eles não reconsideram os agrupamentos previamente formados. Por último, na abordagem centralizada, se utiliza as múltiplas visões de forma simultânea para encontrar padrões escondidos nos dados; representa um desafio importante pois requer uma modificação profunda do processo de agrupamento (CLEUZIOU et al., 2009).

Várias técnicas de agrupamento *multi-view* centralizado foram aplicadas com sucesso em *feature data* (BICKEL; SCHEFFER, 2004; TZORTZIS; LIKAS, 2010). Contudo, ainda há desafios para essas técnicas. Os dados podem não ser facilmente descritos como vetores, o que geralmente ocorre com dados multimídia e por isso podem não ser facilmente comparáveis entre si. Além disso, mesmo quando as visões podem ser expressas como vetores, ainda há dificuldades quando os atributos possuem propriedades estatísticas muito distintas.

Abordagens centralizadas de agrupamento que operam em matrizes relacionais descritas por múltiplas matrizes de dissimilaridade (FRIGUI; HWANG; RHEE, 2007; CARVALHO; LECHEVALLIER; MELO, 2013; CARVALHO; MELO; LECHEVALLIER, 2015) lidam facilmente com essas dificuldades, elas precisam apenas de uma medida de dissimilaridade adequada afim de descrever os relacionamentos entres os objetos de acordo com cada visão. Dados relacionais podem ser muito úteis quando: uma medida de dissimilaridade específica é necessária para resolver um dado problema, confidencialidade é necessária, uma vez que não será necessário ter acesso aos dados em si apenas ao grau de dissimilaridade entre eles, ou quando a natureza dos dados é diferente.

Agrupamento de dados é uma tarefa difícil, especialmente para conjuntos de dados grandes, com alta dimensão e múltiplas fontes. Supervisão parcial, que geralmente está ligado à pertinência de alguns objetos em determinados grupos ou restrições par-a-par (*must-link* e *cannot-link*) entre objetos, podem mitigar esse problema (CHAPELLE; SCHOEKOPF; ZIEN, 2006; GRIRA; CRUCIANU; BOUJEMAA, 2008). Referências (FRIGUI; HWANG, 2008) e (MELO; CARVALHO, 2013) proveem técnicas de agrupamento *multi-view* centralizado que operam em dados relacionais descritos por múltiplas matrizes de dissimilaridade com supervisão parcial par-a-par. Sendo eles o SS-CARD e o SS-MVFCSMdd, respectivamente.

Na supervisão parcial par-a-par objetos que devem pertencer à mesma partição são indicados como *must-link*, enquanto que objetos que devem pertencer à partições diferentes são indicados como *cannot-link*. O algoritmo tentará atender à essas restrições mas elas não são obrigatórias, apenas servem de guia no processo de agrupamento.

1.1 Objetivo

Nesse contexto, o objetivo desse trabalho é a proposição de um algoritmo do tipo *fuzzy c-medoids vectors* semi-supervisionado de dados relacionais representados por múltiplas matrizes de dissimilaridade, chamado daqui em diante de SS-MVFCVSMdd (Semi-Supervised Multi-View Fuzzy Clustering Vector Set-Medoids). O SS-MVFCVSMdd tem por finalidade encontrar uma partição *fuzzy* do conjunto de dados de entrada ao mesmo tempo que encontra um vetor de pesos de relevância para cada matriz de dissimilaridade e um vetor de conjunto de protótipos (*medoids*) para cada grupo.

O algoritmo proposto é bem similar ao SS-MVFCSMdd (MELO; CARVALHO, 2013) mas se diferencia pelo fato de utilizar **vetores** de set-medoids ao passo em que o SS-MVFCSMdd utiliza apenas set-medoids. Isso ocorre devido aos algoritmos em que eles se baseiam, como será discorrido em outras seções deste trabalho.

Consequentemente, em relação ao SS-CARD (FRIGUI; HWANG, 2008) o algoritmo proposto funciona de maneira bem diferente, visto que o SS-CARD é baseado no CARD (FRIGUI; HWANG; RHEE, 2007) que por sua vez utiliza estratégias do NERF (HATHAWAY; BEZDEK, 1994) . Portanto, o SS-CARD não se utiliza de set-medoids, como ocorre no modelo proposto e no SS-MVFCSMdd. Essas diferenças ficarão mais claras na seções que explicam cada um desses algoritmos além de algoritmos relacionados.

No algoritmo proposto o critério de adequação (função objetivo) também leva em consideração restrições par-a-par do tipo *must-link* e *cannot-link* sobre os dados, assim como no SS-CARD e no SS-MVFCSMdd. Essas restrições podem ser construídas a partir dos rótulos dos objetos, o que se caracteriza como uma semi-supervisão.

Esse trabalho também apresenta uma análise dos resultados da aplicação do modelo proposto em experimentos com bases de dados reais que foram extraídas de variadas fontes para avaliar o desempenho desse modelo.

O modelo proposto também foi submetido como artigo e aceito na *IEEE International Conference on Fuzzy Systems 2017*. A carta de aceitação e a primeira página do artigo em questão se encontram no apêndice B.

1.2 Estrutura da Dissertação

Esta dissertação está organizada em 6 capítulos. O presente Capítulo 1 apresentou uma breve introdução a técnicas e métodos de agrupamento de dados, características dos dados utilizados, supervisão parcial par-a-par em agrupamento, objetivo do trabalho e estrutura da dissertação.

Já o Capítulo 2 explica um pouco sobre agrupamento relacional, seleção de atributos em agrupamento, agrupamento semi-supervisionado com restrições par-a-par e apresenta alguns modelos relacionados ao SS-MVFCVSMdd. O Capítulo 3 apresenta o modelo pro-

posto: SS-MVFCVSMdd. O Capítulo 4 descreve os experimentos realizados, delineando a metodologia, métricas e conjuntos de dados utilizados. O Capítulo 5 mostra os resultados dos experimentos, comentários sobre esses resultados e um exemplo de saída do algoritmo proposto. Por fim, o Capítulo 6 encerra com comentários finais e propostas de trabalhos futuros.

2 MODELOS RELACIONADOS

Esta seção descreve vários modelos relacionados ao modelo que será proposto mais adiante. Ao entender esses modelos, a compreensão do modelo proposto no capítulo 3 torna-se trivial.

A figura 1 mostra destacados em negrito os algoritmos relevantes que serão discutidos neste trabalho bem como o algoritmo proposto no capítulo 3, além de mostrar também os algoritmos relacionados que serviram de inspiração para os algoritmos de agrupamento *fuzzy* semi-supervisionado de dados relacionais descritos por múltiplas matrizes de dissimilaridade presentes na figura.

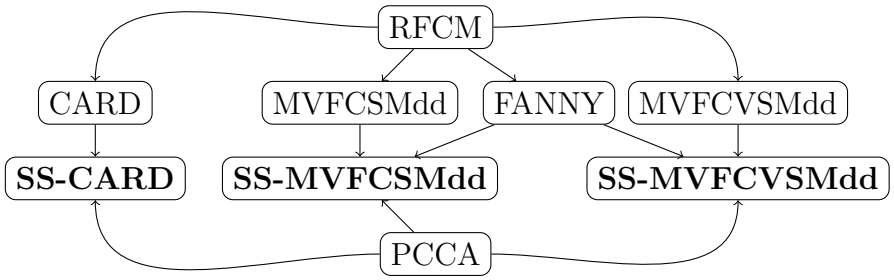

Figura 1 – Grafo directionado mostrando os algoritmos de agrupamento *fuzzy* semi-supervisionado de dados relacionais descritos por múltiplas matrizes de dissimilaridade, destacados em negrito, e algoritmos de agrupamento *fuzzy* relacionados. O arco direcionado (x, y) denota que o algoritmo x teve influência no algoritmo y.

2.1 Agrupamento Relacional

Seja $E = \{e_1, \ldots, e_N\}$ o conjunto de N objetos. Seja $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$ a matriz que representa a partição *fuzzy* de E em C grupos, com a componente $\mathbf{u}_i = (u_{i1}, \ldots, u_{ik}, \ldots, u_{iC})$ sendo o vetor de graus de pertinência do objeto e_i nos grupos *fuzzy*, onde u_{ik} é o grau de pertinência do objeto e_i no grupo *fuzzy* k. Dados relacionais consistem de um conjunto de P matrizes relacionais $(N \times N)$, ou seja $\mathbf{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_P\}$, onde $\mathbf{D}_j[k, l] = d_j(e_k, e_l)$ dá a dissimilaridade entre os objetos e_k e e_l na matriz de dissimilaridade $\mathbf{D}_j$. $\mathbf{D}_j$ está sujeita à:

$$\mathbf{D}_j[k, l] \geq 0 \quad \mathbf{D}_j[k, k] = 0 \quad \mathbf{D}_j[k, l] = \mathbf{D}_j[l, k] \tag{2.1}$$

Alguns algoritmos como o FANNY (ROUSSEEUW; KAUFMAN, 1990), NERF e RFCM (*relational fuzzy c-means*) (HATHAWAY; BEZDEK, 1994), trabalham com apenas uma ma-

triz de (dis)similaridade e levam isso em conta a fim de realizar o agrupamento, enquanto outros, como o CARD (FRIGUI; HWANG; RHEE, 2007), MVFCSMdd (CARVALHO; LECHEVALLIER; MELO, 2013) e o MVFCVSMdd (CARVALHO; MELO; LECHEVALLIER, 2015) podem levar em conta mais de uma matriz simultaneamente.

2.1.1 Agrupamento e Seleção de Visões

Seja $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$ o vetor de relevâncias, onde cada componente $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kP})$ é o vetor de relevância das matrizes de dissimilaridade no grupo k, onde λ_{kj} é o valor de relevância da matriz de dissimilaridade $\mathbf{D}_j$ no grupo *fuzzy* k. Em outras palavras, podemos considerar $\boldsymbol{\Lambda}$ como uma matriz: $\boldsymbol{\Lambda} = [\lambda_{kj}]\,(k = 1, \ldots, C; j = 1, \ldots, p)$.

Como dito anteriormente, alguns algoritmos podem levar em conta múltiplas matrizes de (dis)similaridade. Nesse caso, eles podem inferir pesos de relevância para cada matriz de dissimilaridade e levar isso em conta no processo de agrupamento. Os algoritmos SS-CARD, SS-MVFCSMdd e SS-MVFCVSMdd possuem tal comportamento.

2.1.2 Agrupamento Semi-supervisionado com Restrições par-a-par

Seja $\mathcal{M}$ o conjunto de pares *must-link* tal que $(l, m) \in \mathcal{M}$ implica que os objetos e_l e e_m podem ser designados ao mesmo grupo. Seja $\mathcal{C}$ o conjunto de pares *cannot-link* tal que $(l, m) \in \mathcal{C}$ implica que os objetos e_l e e_m podem ser designados a grupos diferentes.

O agrupamento semi-supervisionado com restrições par-a-par tem como foco melhorar os resultados do agrupamento através da introdução de restrições (*constraints*) que indicam se um par de objetos deve ser designado para um mesmo grupo ou grupos diferentes. Note que essas restrições são *soft*, ou seja, um esforço é feito afim de se satisfazer as restrições mas não é garantido. Claro que essas restrições não devem ser contraditórias (e.g. se $(l, m) \in \mathcal{M}$ então $(l, m) \notin \mathcal{C}$, vise-versa). Essas restrições podem ser dadas como entrada pelo usuário ou podem ser derivadas de uma classificação feita previamente por uma especialista.

2.2 SS-CARD

A abordagem semi-supervisionada para agrupamento e agregação de dados relacionais (SS-CARD) (FRIGUI; HWANG, 2008) é baseado no algoritmo NERF (HATHAWAY; BEZDEK, 1994). Esse modelo integra características de ambos CARD (FRIGUI; HWANG; RHEE, 2007) e *pairwise-constrained competitive agglomeration* (PCCA) (GRIRA; CRUCIANU; BOUJEMAA, 2005) afim de agregar as dissimilaridades das diversas matrizes e aprender um peso de relevância para cada matriz em cada grupo.

O SS-CARD minimiza a seguinte função objetivo apresentada na equação (2.2)

$$J = J^{unsup} + \alpha \times J^{sup} \tag{2.2}$$

onde

$$J^{unsup} = \sum_{k=1}^{C} \frac{\sum_{i=1}^{N} \sum_{h=1}^{N} u_{ik}^2 u_{hk}^2 D_{\lambda_k}(e_i, e_h)}{2 \sum_{h=1}^{N} u_{hk}^2} \tag{2.3}$$

e

$$J^{sup} = \sum_{(l,m)\in\mathcal{M}} \sum_{r=1}^{C} \sum_{\substack{s=1 \\ s\neq r}}^{C} u_{lr} u_{ms} + \sum_{(l,m)\in\mathcal{C}} \sum_{r=1}^{C} u_{lr} u_{mr} \tag{2.4}$$

sendo

$$D_{\lambda_k}(e_i, e_h) = \sum_{j=1}^{P} \lambda_{kj}^q \, d_j(e_i, e_h) \tag{2.5}$$

sujeito à:

$$\lambda_{kj} \in [0, 1] \quad \forall k, j \quad \text{e} \quad \sum_{j=1}^{P} \lambda_{kj} = 1 \quad \forall k \tag{2.6}$$

O primeiro termo de (2.2) é a função objetivo do CARD enquanto que o segundo termo se origina do algoritmo PCCA, que penaliza violações de restrições baseado no grau de pertinência dos objetos. O valor de α controla a importância da informação supervisionada, caso seu valor seja muito pequeno, boa parte da informação supervisionada será ignorada. Por outro lado, se α for muito grande, então algumas restrições seriam forçadas a serem satisfeitas ao custo de afetar a estrutura dos grupos.

Em (FRIGUI; HWANG, 2008) o valor de α é proporcional à razão entre os dois termos da função objetivo do SS-CARD (Equação (2.2)) e se altera a cada iteração do algoritmo:

$$\alpha = \frac{J^{unsup}}{J^{sup}} \tag{2.7}$$

A minimização de J com respeito à matriz pertinência U se dá pela equação (2.8).

$$u_{ik} = u_{ik}^{RFCM} + u_{ik}^{Const} \tag{2.8}$$

sujeito às restrições

$$u_{ik} \geq 0 \quad \forall ik \quad \text{e} \quad \sum_{k=1}^{C} u_{ik} = 1 \quad \forall i \tag{2.9}$$

onde

$$u_{ik}^{RFCM} = \frac{1}{\sum_{r=1}^{C} (a_{ik}/a_{ir})}$$
$$u_{ik}^{Const} = \frac{\alpha}{a_{ik}} (\overline{C_i} - C_{ik})$$

e

$$\begin{aligned} a_{ik} &= 2d_{ik}^2 \\ &= \frac{2 \sum_{h=1}^{N} u_{hk}^2 D_{\lambda_k}(e_i, e_h)}{\sum_{h=1}^{N} u_{hk}^2} - \frac{\sum_{p=1}^{N} \sum_{h=1}^{N} u_{hk}^2 u_{pk}^2 D_{\lambda_k}(e_h, e_p)}{(\sum_{h=1}^{N} u_{hk}^2)^2} \end{aligned} \tag{2.10}$$

e

$$C_{ik} = \sum_{(i,m)\in\mathcal{M}} \sum_{r=1,r\neq k}^{C} u_{mr} + \sum_{(i,m)\in\mathcal{C}} u_{mk}$$
$$\overline{C_i} = \frac{\sum_{k=1}^{C}(\sum_{(i,m)\in\mathcal{M}} \sum_{r=1,r\neq k}^{C} u_{mr} + \sum_{(i,m)\in\mathcal{C}} u_{mk})/a_{ik}}{\sum_{k=1}^{C}(1/a_{ik})}$$

É importante notar a possibilidade de u_{ik} estar fora do intervalo $[0, 1]$. Neste caso, (FRIGUI; HWANG, 2008) propõe um *clipping* desses valores para 0 ou 1 e a renormalização desses valores para que somem 1. Se muitos *clippings* estiverem ocorrendo, então alterar o valor de α pode ser uma alternativa pois ele pode estar muito baixo ou muito alto. Assim como no NERF também é possível que algumas distâncias sejam não-euclideanas no SS-CARD. Para resolver essa questão, a transformação β-*spread* usada no NERF (HATHAWAY; BEZDEK, 1994) é aplicada.

Minimização do J com respeito aos pesos de relevância W se dá através da equação (2.11) sujeito à restrição (2.6).

$$\lambda_{kj} = \frac{1}{\sum_{p=1}^{P}(\overline{D}_{kj}/\overline{D}_{kp})^{1/(q-1)}} \tag{2.11}$$

onde

$$\overline{D}_{kj} = \sum_{i=1}^{N}\sum_{h=1}^{N} u_{ik}^2 u_{hk}^2 d_j(e_i, e_h) \tag{2.12}$$

O algoritmo completo do SS-CARD é mostrado pelo Algoritmo 1. No loop principal, o algoritmo inicia computando as distâncias de acordo com a equação (2.10), então, o algoritmo aplica a transformação β-*spread* caso haja algum $d_{ik}^2(i = 1, \ldots, N; k = 1, \ldots, C)$ negativo. Feito isso, o algoritmo atualiza o valor de α de acordo com a equação (2.7) para que então a matrizes $\mathbf{U}$ e Λ sejam atualizadas de acordo com as equações (2.8) e (2.11) respectivamente e nessa ordem.

O critério de parada é feito através da comparação do valor da função objetivo na iteração atual com o valor da mesma na iteração anterior, caso a diferença absoluta entre elas seja menor que ϵ ou se o número máximo de iterações T tiver ocorrido.

Algoritmo 1: SS-CARD Algorithm

INPUT
  $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_P\}$: the set of P dissimilarity $(N \times N)$ matrices;
  C: the number of clusters
  T: the maximum number of iterations;
  $0 < \epsilon \ll 1$: stopping parameter
  $q \in [1, \text{inf})$: discriminant exponent
  $\mathcal{M}$: the set of must-link constraints
  $\mathcal{C}$: the set of cannot-link constraints
OUTPUT
  the C-dimensional vector of relevance weight vectors $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$
  the N-dimensional vector of membership degree vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$
INITIALIZATION
  Set $t = 0$
  Set $\boldsymbol{\lambda}_k = (1/P, \ldots, 1/P)$, $(1 \leq i \leq C)$
  Randomly set $u_{ik} \in [0, 1]$, with $\sum_{k=1}^{C} u_{ik} = 1$
  Compute J according to equation (2.2)
repeat
  Set $t = t + 1$
  Compute distances using equation (2.10)
  if $d_{ik}^2 < 0$ for some i, k **then**
    Apply the β-spread transform as in NERF
  end if
  Update α using equation (2.7)
  Compute the membership degree u_{ik} of object e_i into the fuzzy cluster k according to equation (2.8)
  Compute the relevance weight λ_{kj} of the dissimilarity matrix $\mathbf{D}_j$ into the fuzzy cluster k according to algorithm (2.11)
  Set $J_{\text{old}} = J$
  Compute J according to equation 2.2
  Set $J_{\text{new}} = J$
until $|J_{\text{new}} - J_{\text{old}}| < \epsilon$ or $t > T$

2.3 MVFCSMdd

(CARVALHO; LECHEVALLIER; MELO, 2013) propuseram o algoritmo iterativo MVFCSMdd (multi-view relational fuzzy c-medoids clustering) baseado no *relational fuzzy c-medoids* (RFCM). Ele possui variações tanto para ponderação local como global das matrizes de dissimilaridade e pode utilizar um dos dois tipos diferentes de restrições em relação à ponderação:

$$\prod_{j=1}^{P} \lambda_{kj} = 1 \, \forall k \in 1, ..., K \tag{2.13}$$

$$\sum_{j=1}^{P} \lambda_{kj} = 1 \, \forall k \in 1, ..., K \tag{2.14}$$

Este trabalho tem foco na variação com ponderação local para as matrizes de dissimilaridade com a restrição (2.13). Seja o conjunto de *medoids* (*set-medoids*) $G_k \, (k = 1, \ldots, C)$ o representante do grupo *fuzzy* k onde cada componente é um subconjunto de cardinalidade fixa $1 \leq q \ll N$ do conjunto de objetos E i.e., $G_k \in E^q = \{A \subset E : |A| = q\}$.

O MVFCSMdd provê:

- uma partição *fuzzy* de E em C grupos representados por um vetor N-dimensional de vetores de grau de pertinência (um para cada objeto) $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$, com a componente $\mathbf{u}_i = (u_{i1}, \ldots, u_{ik}, \ldots, u_{iC})$ sendo o vetor de graus de pertinência do objeto e_i nos grupos *fuzzy*, onde u_{ik} é o grau de pertinência do objeto e_i no grupo *fuzzy* k;

- um vetor C-dimensional de vetores de pesos de relevância $\Lambda = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$, com a componente $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kP})$ sendo o vetor de pesos de relevância das matrizes de dissimilaridade no grupo k, onde λ_{kj} é o peso de relevância da matriz de dissimilaridade $\mathbf{D}_j$ no grupo *fuzzy* k;

- um vetor C-dimensional de conjuntos de *medoids* (um para cada grupo *fuzzy*) $\mathbf{G} = (G_1, \ldots, G_k, \ldots, G_P)$, sendo cada componente o representante do grupo *fuzzy* k.

No algoritmo MVFCSMdd, a partir de uma solução inicial, o vetor $\mathbf{G}$ de *set-medoids*, o vetor $\boldsymbol{\Lambda}$ de vetores de pesos de relevância e o vetor $\mathbf{U}$ de vetores de graus de pertinência são obtidos de forma iterativa em três passos pela minimização da função objetivo J mostrada na equação (2.15).

$$J = \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})^m D_{\boldsymbol{\lambda}_k}(e_i, G_k) \tag{2.15}$$

onde

$$D_{\boldsymbol{\lambda}_k}(e_i, G_k) = \sum_{j=1}^{P} \lambda_{kj} D_j(e_i, G_k) = \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e) \tag{2.16}$$

O parâmetro $m \in]1, \infty[$ controla o grau de fuzzificação da pertinência de cada objeto e_i. O algoritmo alcança a minimização da função objetivo através da execução de três passos iterativamente: busca pelo melhor conjunto de *medoids*, computação do melhor vetor de pesos de relevância e definição da melhor partição *fuzzy*.

2.3.1 Busca pelo melhor conjunto de medoids

Nesse passo, $\lambda_k\,(k = 1, ..., C)$ e a matriz de pertinência U são mantidas fixas enquanto $G_k\,(k = 1, ..., C)$ é atualizado.

O conjunto de *medoids* G_k, representante do grupo k, que minimiza J, é tal que:

$$\sum_{i=1}^{N}(u_{ik})^m \sum_{e \in G^*} d_j(e_i, e) \longrightarrow Min$$

O algoritmo 2 descreve o processo.

Algoritmo 2: Medoid Vector Update Algorithm

1: **for** $k = 1$ to C **do**
2: $G_k \leftarrow \emptyset$
3: **repeat**
4: Find $e_l \in E$ such that:
5: $l = argmin_{1 \leq h \leq N} \sum_{i=1}^{N}(u_{ik})^m \sum_{j=1}^{P} \lambda_{kj}\, d_j(e_i, e_h)$
6: $G_k \leftarrow G_k \cup \{e_l\}$
7: **until** $(|G_k| = q)$
8: **end for**

2.3.2 Computação do melhor vetor de pesos de relevância

Nesse passo o vetor de protótipos (*set-medoids*) $G_k\,(k = 1, ..., C)$ e a matriz de pertinência U são mantidos fixos enquanto que $\boldsymbol{\lambda}_k\,(k = 1, ..., C)$ é atualizado. Seja $V = \{j \in \{1, ..., P\} : \sum_{i=1}^{N}(u_{ik})^m \sum_{e \in g_k} d_j(e_i, e) \leq \theta\}$, para algum $0 < \theta \ll 1$.

Usando o método de multiplicadores de Lagrange com a restrição (2.13) obtemos:

$$\mathcal{L} = J - \sum_{k=1}^{C} \omega_k \left(\left[\prod_{j=1}^{P} \lambda_{kj} \right] - 1 \right) \tag{2.17}$$

Após o ajuste das derivadas parciais de $\mathcal{L}$ com respeito à λ e ω_k obtemos:

$$\lambda_{kj} = \frac{\left\{ \prod_{h=1}^{P} \left[\sum_{i=1}^{N}(u_{ik})^m \sum_{e \in G_k} d_h(e_i, e) \right] \right\}^{\frac{1}{P}}}{\left[\sum_{i=1}^{N}(u_{ik})^m \sum_{e \in G_k} d_j(e_i, e) \right]} \tag{2.18}$$

Dada a equação (2.18), temos que considerar o caso em que $\sum_{i=1}^{N}(u_{ik})^m \sum_{e \in G_k} d_j(e_i, e) = 0$ para algum j. O algoritmo 3 descreve a atualização de $\boldsymbol{\Lambda}$ levando esse caso em conta.

2.3.3 Definição da melhor partição fuzzy

Nesse passo o vetor de *set-medoids* $G_k\,(k = 1, ..., C)$ e o os pesos $\boldsymbol{\lambda}_k\,(k = 1, ..., C)$ são mantidos fixos enquanto que a matrix de pertinência U é atualizada. Seja $A = \{k \in \{1, ..., C\} : D_{\boldsymbol{\lambda}_k}(e_i, G_k) = 0\}$ para um dado objeto i. Usando multiplicadores de Lagrange é possível encontrar que U é computado como exposto no algoritmo 4 sujeito à (2.9).

Algoritmo 3: Relevance Weights Vector Update Algorithm

1: **if** $\forall j \in V$ **then**
2: $\quad \lambda_{kj}$ remains unchanged
3: **else if** $\forall j \notin V$ **then**
4: $\quad \chi \leftarrow \frac{1}{\prod_{j \in V} \lambda_{kj}}$
5: $\quad \lambda_{kj} \leftarrow \frac{\left\{\chi \times \prod_{h \notin V} \left[\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_k} d_h(e_i, e)\right]\right\}^{\frac{1}{P-|V|}}}{\left[\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e)\right]}$
6: **end if**

Algoritmo 4: Fuzzy Membership Update Algorithm

1: **if** $A \neq \emptyset$ **then**
2: $\quad u_{ik} = 1/|A| \quad , \forall k \in A$
3: $\quad u_{ir} = 0 \quad , \forall r \notin A$
4: **else if** $A = \emptyset$ **then**
5: $\quad u_{ik} = \left[\sum_{r=1}^{C} \left(\frac{D_{\boldsymbol{\lambda}_k}(e_i, G_k)}{D_{\boldsymbol{\lambda}_r}(e_i, G_r)}\right)^{\frac{1}{m-1}}\right]^{-1}$
6: **end if**

O algoritmo completo do MVFCSMdd é descrito pelo algoritmo 5. O loop principal do algoritmo parte de uma partição *fuzzy* inicial e alterna entre os três seguintes passos: busca pelo melhor conjunto de medoids, computação do melhor vetor de pesos de relevância e definição da melhor partição *fuzzy*. Até atingir convergência, seja por um valor estacionário da função objetivo ou por atingir o número máximo de iterações.

Algoritmo 5: MVFCSMdd Algorithm

INPUT
  $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_P\}$: the set of P dissimilarity $(N \times N)$ matrices;
  C: the number of cluster
  T: the maximum number of iterations;
  $m \in [1, \infty)$: mebership fuzziness parameter
  $0 < \epsilon \ll 1$: stopping parameter
  $q_j \, (j = 1, \ldots, P)$: cardinal of the set-medoids
OUTPUT
  the C-dimensional vector of set-medoids $\mathbf{G} = (G_1, \ldots, G_k, \ldots, G_C)$
  the C-dimensional vector of relevance weight vectors $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$
  the N-dimensional vector of membership degree vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$
INITIALIZATION
  Set $t = 0$
  Randomly select C distinct vectors of set-medoids $G_k \in E^{(q)} \, (1 \leq k \leq C)$
  Set $\boldsymbol{\lambda}_k = (1, \ldots, 1)$, $(1 \leq k \leq C)$
  Randomly set $u_{ik} \in [0, 1]$, with $\sum_{k=1}^{C} u_{ik} = 1$
  Compute J according to equation (2.15)
repeat
  Set $t = t + 1$
  Representation step
    Compute the set-medoids G_k according to algorithm 2
  Weighting step
    Compute the relevance weight λ_{kj} of the dissimilarity matrix $\mathbf{D}_j$ into the fuzzy cluster k according to algorithm (3)
  Allocation step
    Compute the membership degree u_{ik} of object e_i into the fuzzy cluster k according to algorithm 4
  Set $J_{\text{old}} = J$
  Compute J according to equation (2.15)
  Set $J_{\text{new}} = J$
until $|J_{\text{new}} - J_{\text{old}}| < \epsilon$ or $t > T$

2.4 SS-MVFCSMdd

SS-MVFCSMdd (MELO; CARVALHO, 2013) é um algoritmo de agrupamento *fuzzy c-medoids* semi-supervisionado de dados relacionais baseado no MVFCSMdd (Seção 2.3) a fim de produzir uma partição *fuzzy* consenso dos dados combinado à características do algoritmo PCCA (GRIRA; CRUCIANU; BOUJEMAA, 2005), que introduz restrições para-a-par *must-link* e *must-not-link* afim de obter-se um melhor processo de agrupamento. Considere $G_k\,(k = 1, \ldots, C)$ como definido na seção 2.3.

O SS-MVFCSMdd minimiza a função objetivo dada pela equação (2.19), similar à equação (2.2):

$$J = J^{unsup} + \alpha \times J^{sup} \tag{2.19}$$

Onde J^{unsup} é idêntico à função objetivo do MVFCSMdd (equação (2.15)) como mostra a equação (2.20).

$$J^{unsup} = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^m D_{\boldsymbol{\lambda}_k}(e_i, G_k) \tag{2.20}$$

Onde $D_{\boldsymbol{\lambda}_k}(e_i, G_k)$ é descrito pela equação (2.16). Além disso, há também a adição de um segundo termo que leva em conta a violação das restrições, no mesmo estilo da função objetivo do SS-CARD, portanto J^{sup} é o mesmo que na equação (2.4). O SS-MVFCSMdd alcança a minimização da sua função objetivo através da execução dos mesmos três passos feitos pelo MVFCSMdd. Os primeiros dois passos, busca pelo melhor conjunto de *medoids* e computação do melhor vetor de pesos de relevância, são exatamente os mesmos para ambos modelos. Para este algoritmo, o parâmetro m **deve** ser igual à 2.0.

Contudo a definição da melhor partição *fuzzy* para o SS-MVFCSMdd é diferente, uma vez que as restrições são levadas em conta. Utilizando o método de multiplicadores de Lagrange levando em conta as restrições (2.9) obtem-se:

$$\mathcal{L} = J - \sum_{i=1}^{N}\gamma_i\left(\left[\sum_{k=1}^{C}u_{ik}\right] - 1\right) - \sum_{k=1}^{C}\sum_{i=1}^{N}\psi_{ik}u_{ik} \tag{2.21}$$

Afim de garantir valores não-negativos para os graus de pertinência e levando em conta a função objetivo (2.19) e as restrições (2.9) as condições de Karush-Kuhn-Tucker (KKT) correspondentes são:

$$\begin{aligned} \psi_{ik} &\geq 0 \\ u_{ki}\psi_{ki} &= 0 \\ \frac{\partial \mathcal{L}}{\partial u_{ki}} &= u_{ki}a_{ki} + b_{ki} - \gamma_k - \psi_{ki} = 0 \end{aligned} \tag{2.22}$$

onde:

$$
\begin{aligned}
a_{ik} &= D_{\boldsymbol{\lambda}_k}(e_i, G_k) = 2 \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e) \\
b_{ik} &= 2\,\alpha \left(\sum_{(e_i,e_m) \in \mathcal{M}} \sum_{\substack{s=1 \\ s \neq k}}^{C} u_{ms} + \sum_{(e_i,e_m) \in \mathcal{C}} u_{mk} \right)
\end{aligned}
\tag{2.23}
$$

Uma solução algorítmica pode ser obtida através da combinação das relações (2.9) e (2.22) de forma similar ao FANNY. A solução é exposta no algoritmo 6 que realiza a etapa de definição da melhor partição *fuzzy* para o SS-MVFCSMdd.

O algoritmo completo do SS-MVFCSMdd é descrito pelo Algoritmo 7. Assim como no MVFCSMdd, esse algoritmo parte de uma partição *fuzzy* inicial e alterna entre os três seguintes passos: busca pelo melhor conjunto de medoids, definido na Subseção 2.3.1; computação do melhor vetor de pesos de relevância, definido na Subseção 2.3.2; e definição da melhor partição *fuzzy*, definido na Seção 2.4 descrito pelo algoritmo 6. Até atingir convergência, seja por um valor estacionário da função objetivo ou por atingir o número máximo de iterações.

2.4.1 Escolha do α

O valor de α controla a importância da informação supervisionada, caso seu valor seja muito pequeno, boa parte da informação supervisionada será ignorada. Por outro lado, se α for muito grande, então algumas restrições seriam forçadas a serem satisfeitas ao custo de afetar a estrutura dos grupos.

A abordagem proposta no SS-CARD (seção 2.2) apresenta possíveis problemas. Essa abordagem viola a garantia de minimização da função objetivo visto que o α muda, com certa arbitrariedade, em cada iteração; além disso os valores de α escolhidos podem não ser suficientes para que as restrições sejam consideradas.

Referência (MELO; CARVALHO, 2013) propõem um método simples, mas que não quebra a garantia da convergência da função objetivo, para escolha do α. Eles propõem que o valor utilizado seja fixo e para fazer a escolha do seu valor considera que a informação das restrições par-a-par estão corretas e devem ser refletidas na partição *fuzzy* final. Ou seja, o α escolhido faz com que o termo da equação (2.4) apresente um valor, idealmente, próximo de zero quando computado utilizando a partição final.

A proposta de Melo e Carvalho é descrita pelo algoritmo 8. α pode assumir qualquer valor $(2n+1)\alpha_{\min}$, tal que $n \geq 0$, desde que esteja no intervalo $[\alpha_{\min}, \alpha_{\max}]$. Como se pode observar, o número de máximo de atualizações de α é de $\mathcal{O}(log(\alpha_{\max}))$.

O valor inicial de α (denotado por $\alpha_{\min}$) deve partir, idealmente, de um valor que não seja grande. Em (MELO; CARVALHO, 2013) o valor inicial e máximo de α escolhidos foram,

Algoritmo 6: SS-MVFCSMdd Fuzzy Membership Update

for $i = 1$ to N **do**
  $A_i = \emptyset$
  $A_i = \{k \in \{1, \ldots, C\} : \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e) = 0\}$
  if $A_i \neq \emptyset$ **then**
    $u_{ik} = \frac{1}{|A_i|}, \forall k \in A_i$
    $u_{ir} = 0, \forall r \notin A_i$
  else
    $V_i = \{1, \ldots, C\}$
    for $k = 1$ to C **do**
      $a_{ik} = 2 \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_k} d_j(e_i, e)$;
      $b_{ik} = 2\,\alpha \left(\sum_{(e_i, e_m) \in \mathcal{M}} \sum_{\substack{s=1 \\ s \neq k}}^{C} u_{ms} + \sum_{(e_i, e_m) \in \mathcal{C}} u_{mk} \right)$
    end for
    repeat
      $test = 0$
      $\gamma_i = \frac{1 + \sum_{w \in V_i} \frac{b_{iw}}{a_{iw}}}{\sum_{w \in V_i} \frac{1}{a_{iw}}}$;
      for $k = 1$ to C **do**
        if $k \in V_i$ **then**
          $u_{ik} = \frac{\gamma_i - b_{ik}}{a_{ik}}$
          if $u_{ik} \leq 0$ **then**
            $u_{ik} = 0$;
            $V_i = V_i \backslash \{k\}$;
            $test = 1$;
          end if
        end if
      end for
    until $test \neq 1$
  end if
end for

respectivamente, 1 e 100, garantindo que o valor será encontrado em até 7 iterações. A razão da escolha desses valores foi a minimização do tempo de execução em virtude do grande número de experimentos utilizados.

Algoritmo 7: SS-MVFCSMdd Algorithm

INPUT
- $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_P\}$: the set of P dissimilarity $(N \times N)$ matrices;
- C: the number of cluster
- T: the maximum number of iterations
- $m = 2$: membership fuzziness parameter
- $0 < \epsilon \ll 1$: stopping parameter
- $q_j \, (j = 1, \ldots, P)$: cardinal of the set-medoids
- $\alpha > 0$: parameter related to the importance of the supervision (constraints)
- $\mathcal{M}$: the set of must-link constraints
- $\mathcal{C}$: the set of cannot-link constraints

OUTPUT
- the C-dimensional vector of set-medoids $\mathbf{G} = (G_1, \ldots, G_k, \ldots, G_C)$
- the C-dimensional vector of relevance weight vectors $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$
- the N-dimensional vector of membership degree vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$

INITIALIZATION
- Set $t = 0$
- Randomly select C distinct vectors of set-medoids $\mathbf{G}_k \in E^{(q_1)} \times \ldots \times E^{(q_P)}$, $(1 \leq i \leq C)$
- Set $\boldsymbol{\lambda}_k = (1, \ldots, 1)$, $(1 \leq i \leq C)$
- Randomly set $u_{ik} \in [0, 1]$, with $\sum_{k=1}^{C} u_{ik} = 1$
- Compute J according to equation (2.19)

repeat
- Set $t = t + 1$
- **Representation step**
  - Compute the set-medoids G_{kj}, j-th component of the vector of set-medoids $\mathbf{G}_k$ according to algorithm 2
- **Weighting step**
  - Compute the relevance weight λ_{kj} of the dissimilarity matrix $\mathbf{D}_j$ into the fuzzy cluster k according to algorithm (3)
- **Allocation step**
  - Compute the membership degree u_{ik} of object e_i into the fuzzy cluster k according to algorithm 6
- Set $J^{\text{old}} = J$
- Compute J according to equation (2.19)
- $SetJ^{\text{new}} = J$

until $|J^{\text{new}} - J^{\text{old}}| < \epsilon$ or $t > T$

Algoritmo 8: α search algorithm

INPUT
    $\alpha_{\min}$: the initial value of α
    $\alpha_{\max}$: the maximum value of α, $\alpha_{\max} \geq \alpha_{\min}$
OUTPUT
    the final value of α
INITIALIZATION
    $\alpha = \alpha_{\min}$
    $found = false$
repeat
  Execute algorithm 7
  Compute J^{sup} according to equation (2.4) using the final fuzzy partition U
  if $\alpha \geq \alpha_{\max}$ **or** $J^{sup} \leq \epsilon$ **then**
    $found = true$
  else
    $\alpha = \alpha \times 2$
  end if
until $found = true$

3 MODELO PROPOSTO

Este capítulo descreve o modelo proposto SS-MVFCVSMdd e o modelo em que foi baseado, o MVFCVSMdd.

3.1 MVFCVSMdd

Assim como o MVFCSMdd, o MVFCVSMdd (CARVALHO; MELO; LECHEVALLIER, 2015) é um algoritmo de agrupamento *fuzzy* de vetor de *c-medoids* de dados relacionais afim de produzir uma partição *fuzzy* consenso dos dados. Ele também possui variações tanto para ponderação local como global das matrizes de dissimilaridade bem como diferentes restrições para essas ponderações representadas pelas equações (2.14) e (2.13). Este trabalho tem como foco a variação com ponderação local das matrizes de dissimilaridade sujeitas à restrição (2.13).

Seja o vetor de conjunto de *medoids* (*set-medoids*) $\mathbf{G}_k = (G_{k1}, \ldots, G_{kj}, \ldots, G_{kP})\,(k = 1, \ldots, C)$ o representante do grupo *fuzzy* k onde cada componente é um subconjunto de cardinalidade fixa $1 \leq q_j \ll N$ do conjunto de objetos E i.e., $G_{kj} \in E^{q_j} = \{A \subset E : |A| = q_j\}$.

O MVFCVSMdd provê:

- uma partição *fuzzy* de E em C grupos representados por um vetor N-dimensional de vetores de grau de pertinência (um para cada objeto) $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$, com a componente $\mathbf{u}_i = (u_{i1}, \ldots, u_{ik}, \ldots, u_{iC})$ sendo o vetor de graus de pertinência do objeto e_i nos grupos *fuzzy*, onde u_{ik} é o grau de pertinência do objeto e_i no grupo *fuzzy* k;

- um vetor C-dimensional de pesos de relevância (um para cada grupo *fuzzy*) $\Lambda = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$, com a componente $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kP})$ sendo o vetor de pesos de relevância das matrizes de dissimilaridade no grupo k, onde λ_{kj} é o peso de relevância da matriz de dissimilaridade $\mathbf{D}_j$ no grupo *fuzzy* k. Podemos considerar $\boldsymbol{\Lambda}$ como uma matriz: $\boldsymbol{\Lambda} = [\lambda_{kj}]\,(k = 1, \ldots, C; j = 1, \ldots, p)$;

- um vetor C-dimensional de vetores de *set-medoids* $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_k, \ldots, \mathbf{G}_C)$, com cada componente $\mathbf{G}_k = (G_{k1}, \ldots, G_{kj}, \ldots, G_{kP})$ sendo o representante do grupo *fuzzy* k, onde G_{kj} é a componente do representante do grupo *fuzzy* k na matriz de dissimilaridade $\mathbf{D}_j$.

No algoritmo MVFCVSMdd, a partir de uma solução inicial, o vetor $\mathbf{G}$ de vetores de *set-medoids*, o vetor $\boldsymbol{\Lambda}$ de vetores de pesos de relevância e o vetor $\mathbf{U}$ de vetores de graus

de pertinência são obtidos de forma iterativa em três passos pela minimização da função objetivo J mostrada na equação (3.1).

$$J = \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})^m D_{\boldsymbol{\lambda}_k}(e_i, G_k) \tag{3.1}$$

onde

$$D_{\boldsymbol{\lambda}_k}(e_i, G_k) = \sum_{j=1}^{P} \lambda_{kj} D_j(e_i, G_{kj}) = \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_{kj}} d_j(e_i, e) \tag{3.2}$$

O parametro $m \in]1, \infty[$ controla o grau de fuzzificação da pertinência de cada objeto e_i. Os três passos executados iterativamente pelo algoritmo são: busca pelo melhor vetor de *medoids*, computação do melhor vetor de pesos de relevância e definição da melhor partição *fuzzy*.

3.1.1 Busca pelo melhor vetor de medoids

Nesse passo, $\lambda_k \, (k = 1, ..., C)$ e a matriz de pertinência U são mantidas fixas enquanto $G_k \, (k = 1, ..., C)$ é atualizado. O vetor de *set-medoids* $G_k = (G_{k1}, \ldots, G_{kP})$, representante da partição k, que minimiza J, é tal que:

$$\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_j^*} d_j(e_i, e) \longrightarrow Min$$

O algoritmo 9 descreve o processo.

Algoritmo 9: Medoid Vector Update Algorithm

```
1: for k = 1 to C do
2:   for j = 1 to P do
3:     G_kj ← ∅;
4:     repeat
5:       Find e_l ∈ E such that:
6:         l = argmin_{1≤h≤N} Σ_{i=1}^{N} (u_ik)^m d_j(e_i, e_h);
7:       G_kj ← G_k ∪ e_l;
8:     until (|G_kj| = q_j|);
9:   end for
10: end for
```

3.1.2 Computação do melhor vetor de pesos de relevância

Nesse passo, o vetor de *set-medoids* $\mathbf{G}_k \, (k = 1, \ldots, C)$ e a matriz de pertinência $\mathbf{U}$ são mantidos fixos enquanto que $\boldsymbol{\lambda}_k \, (k = 1, \ldots, C)$ é atualizado. Dado que $G_k \, (k = 1, \ldots, C)$

e **U** são mantidos fixos, podemos re-escrever o critério J como descrito pela equação (3.3).

$$J(\lambda_1, \ldots, \lambda_C) = \sum_{k=1}^{C} J_k(\lambda_k) \quad \text{com} \quad J_k(\lambda_k) = J_k(\lambda_{k1}, \ldots, \lambda_{kP}) = \sum_{j=1}^{P} \lambda_{kj} J_{kj} \tag{3.3}$$

onde $J_{kj} = \sum_{i=1}^{N} (u_{ik})^2 D_j(e_i, \mathbf{G}_k) = \sum_{i=1}^{N} (u_{ik})^2 \sum_{e \in \mathbf{G}_k} d_j(e_i, e)$.

Seja $g(\lambda_{k1}, \ldots, \lambda_{kP}) = \lambda_{k1} \times \ldots \times \lambda_{kP} - 1$. Queremos determinar os extremos de $J_k(\lambda_{k1}, \ldots, \lambda_{kP})$ com a restrição $g(\lambda_{k1}, \ldots, \lambda_{kP}) = 0$. Usando o método de multiplicadores de Lagrange com a restrição (2.13) obtemos a equação (2.17) (para $j = 1, \ldots, P$).

Primeiro calculamos as derivadas parciais de $\mathcal{L}$ com respeito à λ_{kj}. Igualando essas derivadas a zero obtemos:

$$\lambda_{kj} = \frac{\left\{ \prod_{h=1}^{P} \left[\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_{kh}} d_h(e_i, e) \right] \right\}^{\frac{1}{P}}}{\left[\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_{kj}} d_j(e_i, e) \right]} \tag{3.4}$$

Dada a equação (3.4), temos que considerar o caso em que $\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_k} d_j(e_i, e) = 0$ para algum j para evitar indeterminação. Seja $V = \{j \in \{1, ..., P\} : \sum_{i=1}^{N} (u_{ik})^m \sum_{e \in g_k} d_j(e_i, e) \leq \theta\}$ com $\theta \to 0$; o algoritmo 10 descreve a atualização de $\mathbf{\Lambda}$ levando esse caso em conta.

Algoritmo 10: Relevance Weights Vector Update Algorithm

1: **if** $\forall j \in V$ **then**
2: $\quad \lambda_{kj}$ remains unchanged
3: **else if** $\forall j \notin V$ **then**
4: $\quad \chi \leftarrow \frac{1}{\prod_{j \in V} \lambda_{kj}}$
5: $\quad \lambda_{kj} \leftarrow \frac{\left\{ \chi \times \prod_{h \notin V} \left[\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_{kh}} d_h(e_i, e) \right] \right\}^{\frac{1}{P - |V|}}}{\left[\sum_{i=1}^{N} (u_{ik})^m \sum_{e \in G_{kj}} d_j(e_i, e) \right]}$
6: **end if**

3.1.3 Definição da melhor partição fuzzy

Nesse passo o vetor de vetores de protótipos $\mathbf{G}_k$ $(k = 1, ..., C)$ e o os pesos $\boldsymbol{\lambda}_k$ $(k = 1, ..., C)$ são mantidos fixos enquanto que a matrix de pertinência U é atualizada. Seja $A = \{k \in \{1, ..., C\} : D_{\boldsymbol{\lambda}_k}(e_i, G_k) = 0\}$. Usando o método dos multiplicadores de Lagrange é possível encontrar que U é computado como exposto no algoritmo 11 sujeito à (2.9).

O algoritmo completo do MVFCVSMdd é descrito pelo algoritmo 12. O loop principal do algoritmo parte de uma partição *fuzzy* inicial e alterna entre os três seguintes passos: busca pelo melhor conjunto de medoids, computação do melhor vetor de pesos de relevância e definição da melhor partição *fuzzy*. Até atingir convergência, seja por um valor estacionário da função objetivo ou por atingir o número máximo de iterações.

Algoritmo 11: Fuzzy Membership Update Algorithm

if $A \neq \emptyset$ **then**
  $u_{ik} = 1/|A| \quad , \forall k \in A$
  $u_{ir} = 0 \quad , \forall r \notin A$
else if $A = \emptyset$ **then**
  $u_{ik} = \left[\sum_{r=1}^{C} \left(\frac{D_{\boldsymbol{\lambda}_k}(e_i, G_{kj})}{D_{\boldsymbol{\lambda}_r}(e_i, G_{rj})} \right)^{\frac{1}{m-1}} \right]^{-1}$
end if

Algoritmo 12: MVFCVSMdd Algorithm

INPUT
  $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_P\}$: the set of P dissimilarity $(N \times N)$ matrices;
  C: the number of cluster
  T: the maximum number of iterations;
  $m \in [1, \infty)$: mebership fuzziness parameter
  $0 < \epsilon \ll 1$: stopping parameter
  $q_j \, (j = 1, \ldots, P)$: cardinal of the set-medoids
OUTPUT
  the C-dimensional vector of set-medoids $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_k, \ldots, \mathbf{G}_C)$
  the C-dimensional vector of relevance weight vectors $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$
  the N-dimensional vector of membership degree vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$
INITIALIZATION
  Set $t = 0$
  Randomly select C distinct vectors of set-medoids $\mathbf{G}_k \in E^{(q_1)} \times \ldots \times E^{(q_P)}$, $(1 \leq i \leq C)$
  Set $\boldsymbol{\lambda}_k = (1, \ldots, 1)$, $(1 \leq i \leq C)$
  Randomly set $u_{ik} \in [0, 1]$, with $\sum_{k=1}^{C} u_{ik} = 1$
  Compute J according to equation (2.15)
repeat
  Set $t = t + 1$
  Representation step
    Compute the set-medoids G_{kj}, j-th component of the vector of set-medoids $\mathbf{G}_k$ according to algorithm 2
  Weighting step
    Compute the relevance weight λ_{kj} of the dissimilarity matrix $\mathbf{D}_j$ into the fuzzy cluster k according to algorithm (3)
  Allocation step
    Compute the membership degree u_{ik} of object e_i into the fuzzy cluster k according to algorithm 4
  Set $JOLD = J$
  Compute J according to equation (2.15)
  Set $JNEW = J$
until $|JNEW - JOLD| < \epsilon$ or $t > T$

3.2 SS-MVFCVSMdd

Assim como o SS-MVFCSMdd introduziu restrições par-a-par ao MVFCSMdd na Seção 2.4, o SS-MVFCVSMdd introduz restrições par-a-par ao MVFCVSMdd. O SS-MVFCVSMdd é um algoritmo semi-supervisionado do tipo *fuzzy c-medoids* para agrupamento de dados relacionais *multi-view*, combinando características do algoritmo MVFCVSMdd e do algoritmo PCCA (GRIRA; CRUCIANU; BOUJEMAA, 2005). Assim como no SS-MVFCSMdd, o algoritmo descrito nesta seção se utilizou das restrições par-a-par *must-link* e *cannot-link* utilizadas no PCCA.

O SS-MVFCVSMdd provê:

- uma partição *fuzzy* de E em C grupos representados por um vetor N-dimensional de vetores de grau de pertinência (um para cada objeto) $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$, com a componente $\mathbf{u}_i = (u_{i1}, \ldots, u_{ik}, \ldots, u_{iC})$ sendo o vetor de graus de pertinência do objeto e_i nos grupos *fuzzy*, onde u_{ik} é o grau de pertinência do objeto e_i no grupo *fuzzy* k;

- um vetor C-dimensional de pesos de relevância (um para cada grupo *fuzzy*) $\Lambda = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$, com a componente $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kj}, \ldots, \lambda_{kP})$ sendo o vetor de pesos de relevância das matrizes de dissimilaridade no grupo k, onde λ_{kj} é o peso de relevância da matriz de dissimilaridade $\mathbf{D}_j$ no grupo *fuzzy* k. Podemos considerar $\boldsymbol{\Lambda}$ como uma matriz: $\boldsymbol{\Lambda} = [\lambda_{kj}]\,(k = 1, \ldots, C; j = 1, \ldots, p)$;

- um vetor C-dimensional de vetores de *set-medoids* (um para cada grupo *fuzzy*) $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_k, \ldots, \mathbf{G}_C)$, com cada componente $\mathbf{G}_k = (G_{k1}, \ldots, G_{kj}, \ldots, G_{kP})$ sendo o representante do grupo *fuzzy* k, onde G_{kj} é a componente do representante do grupo *fuzzy* k na matriz de dissimilaridade $\mathbf{D}_j$.

No algoritmo SS-MVFCVSMdd, a partir de uma solução inicial, o vetor $\mathbf{G}$ de vetores de *set-medoids*, o vetor $\boldsymbol{\Lambda}$ de vetores de pesos de relevância e o vetor $\mathbf{U}$ de vetores de graus de pertinência são obtidos de forma iterativa em três passos pela minimização da função objetivo J mostrada na equação (3.5).

$$J = J^{unsup} + \alpha \times J^{sup} \tag{3.5}$$

Onde J^{sup} é idêntico à função objetivo do MVFCVSMdd (equação (3.1)) como mostra a equação (3.6).

$$J^{unsup} = \sum_{k=1}^{C} \sum_{i=1}^{N} (u_{ik})^2 D_{\boldsymbol{\lambda}_k}(e_i, G_k) \tag{3.6}$$

Onde $D_{\boldsymbol{\lambda}_k}(e_i, G_k)$ é descrito pela equação (3.2). Assim como no SS-CARD e no SS-MVFCSMdd, a função objetivo do SS-MVFCVSMdd também possui uma parte que

trata da violação das restrições, a saber, J^{sup} que é descrito pela equação (2.4). O SS-MVFCVSMdd alcança a minimização da sua função objetivo através da execução dos mesmo três passos feitos pelo MVFCVSMdd. Os primeiros dois passos, busca pelo melhor vetor de *set-medoids* e computação do melhor vetor de pesos de relevância, são exatamente os mesmos para ambos modelos.

Contudo a definição da melhor partição *fuzzy* para o SS-MVFCVSMdd é diferente, uma vez que as restrições *must-link* e *cannot-link* devem ser levadas em conta. A equação (3.7) mostra a fórmula para a definição do grau de pertinência de um dado objeto i em um determinado grupo k.

$$u_{ik} = u_{ik}^{unsup} + u_{ik}^{sup} \tag{3.7}$$

Nota-se que a equação (3.7) possui dois termos, sendo o primeiro pertinente à parte não-supervisionada (u_{ik}^{unsup}), advinda do MVFCVSMdd, e a parte supervisionada (u_{ik}^{sup}) que trata as restrições *must-link* e *cannot-link*, advinda do PCCA.

O termo não-supervisionado é idêntico ao termo do MVFCVSMdd nesta etapa (Subseção 3.1.3), descrito pela equação (3.8). Já o termo supervisionado que trata das restrições *must-link* e *cannot-link* é descrito pelas equações (3.9), (3.10) e (3.11). O que é importante notar na parte supervisionada é a equação (3.10) que tem um papel central de tratar as restrições *must-link* e *cannot-link*; os graus de pertinência de um determinado objeto serão maiores se as restrições *must-link* e *cannot-link* quem dizem respeito a esse objeto forem atendidas.

$$u_{ik}^{unsup} = \left[\sum_{h=1}^{C}\left(\frac{D_{\boldsymbol{\lambda}_k}(e_i, \mathbf{G}_k)}{D_{\boldsymbol{\lambda}_h}(e_i, \mathbf{G}_h)}\right)\right]^{-1} = \left[\sum_{h=1}^{C}\left(\frac{\sum\limits_{j=1}^{p}\lambda_{kj}\sum\limits_{e\in G_{kj}} d_j(e_i, e)}{\sum\limits_{j=1}^{p}\lambda_{hj}\sum\limits_{e\in G_{hj}} d_j(e_i, e)}\right)\right]^{-1} \tag{3.8}$$

$$u_{ik}^{sup} = \frac{\alpha\left(\bar{C}_i - C_{ik}\right)}{2\,D_{\boldsymbol{\lambda}_k}(e_i, G_k)} \tag{3.9}$$

$$C_{ik} = \left(\sum_{(e_i,e_m)\in\mathcal{M}}\sum_{\substack{s=1\\ s\neq k}}^{C} u_{ms} + \sum_{(e_i,e_m)\in\mathcal{C}} u_{mk}\right) \tag{3.10}$$

$$\bar{C}_i = \frac{\sum\limits_{h=1}^{C}\frac{C_{ih}}{2\,D_{\boldsymbol{\lambda}_h}(e_i,G_h)}}{\sum_{h=1}^{C}\frac{1}{2\,D_{\boldsymbol{\lambda}_h}(e_i,G_h)}} \tag{3.11}$$

Contudo, utilizar a equação (3.7) para a definição da melhor partição *fuzzy* pode causar violação das restrições da matriz de pertinência (equação (2.9)). Para um dado objeto i e um dado grupo k, se $\bar{C}_i$ for muito menor que C_{ik} e $D_{\boldsymbol{\lambda}_k}(e_i, G_k)$ for muito pequeno u_{ik} pode se tornar negativo. Consequentemente algum u_{ir}, com $r \neq k$, pode se tornar maior que 1.

Uma maneira de abordar esse problema seria utilizar a abordagem proposta no SS-CARD (FRIGUI; HWANG, 2008): para os objetos em que a violação da restrição (2.9) ocorre, mudar os valores dos graus de pertinência para 0 e 1 e renormalizar os valores para que a soma deles seja igual à 1.

Uma outra forma de abordar esse problema, que foi a utilizada neste trabalho, seria utilizando o método de multiplicadores de Lagrange levando em conta condições Karush-Kuhn-Tucker (KKT) assim como no algoritmo FANNY (ROUSSEEUW; KAUFMAN, 1990).

Utilizando o método de multiplicadores de Lagrange levando em conta as restrições (2.9) obtem-se:

$$\mathcal{L} = J - \sum_{i=1}^{N} \gamma_i \left(\left[\sum_{k=1}^{C} u_{ik} \right] - 1 \right) - \sum_{k=1}^{C} \sum_{i=1}^{N} \psi_{ik} u_{ik} \tag{3.12}$$

Afim de garantir valores não-negativos para os graus de pertinência e levando em conta a função objetivo (3.5) e as restrições (2.9) as condições KKT correspondentes são:

$$\begin{aligned} \psi_{ik} &\geq 0 \\ u_{ki}\psi_{ki} &= 0 \\ \frac{\partial \mathcal{L}}{\partial u_{ki}} &= u_{ki}a_{ki} + b_{ki} - \gamma_k - \psi_{ki} = 0 \end{aligned} \tag{3.13}$$

onde:

$$\begin{aligned} a_{ik} &= D_{\boldsymbol{\lambda}_k}(e_i, G_k) = 2 \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_{kj}} d_j(e_i, e) \\ b_{ik} &= 2\alpha \left(\sum_{(e_i,e_m) \in \mathcal{M}} \sum_{\substack{s=1 \\ s \neq k}}^{C} u_{ms} + \sum_{(e_i,e_m) \in \mathcal{C}} u_{mk} \right) \end{aligned} \tag{3.14}$$

Uma solução algorítmica pode ser obtida através da combinação das relações (2.9) e (3.13) de forma similar ao FANNY. A solução é exposta no algoritmo 13 que realiza a etapa de definição da melhor partição *fuzzy* para o SS-MVFCVSMdd. Mais detalhes na derivação das equações da etapa de definição da melhor partição *fuzzy* encontram-se no apêndice A.

O algoritmo completo do SS-MVFCVSMdd é descrito pelo algoritmo 14. O loop principal do algoritmo parte de uma partição *fuzzy* inicial e alterna entre os três seguintes passos: busca pelo melhor conjunto de medoids, definido na Subseção 3.1.1; computação do melhor vetor de pesos de relevância, definido na Subseção 3.1.2; e definição da melhor partição *fuzzy*, definido na Seção 3.2 descrito pelo algoritmo 13. Até atingir convergência, seja por um valor estacionário da função objetivo ou por atingir o número máximo de iterações.

A complexidade do algoritmo é $\mathcal{O}(CN^2P^2q)$ em cada iteração, considerando que todos os componentes de $\mathbf{G}_k$ possuem a mesma cardinalidade q, ou seja $G_{kj} \in E^q = \{A \subset E : |A| = q\}$. Isso porque é necessário testar cada indivíduo como um possível membro no conjunto de *medoids* e ao mesmo tempo consultar todas as matrizes de dissimilaridade isso para cada uma das componentes de $\mathbf{G_k}$.

A diferença principal em relação ao SS-MVFCSMdd está nesse tratamento de $\mathbf{G}$ como um vetor de vetores de conjunto de *medoids*. É isso que caracteriza a diferença de complexidade entre o algoritmo proposto e o SS-MVFCSMdd.

3.2.1 Escolha do α

Para este trabalho, optamos por utilizar a mesma estratégia adotada em (MELO; CARVALHO, 2013) descrita na Subseção 2.4.1. Ela pode ser facilmente aplicada ao modelo aqui proposto, sem a necessidade de qualquer alteração tanto no modelo quanto na estratégia. Isso porque o SS-MVFCVSMdd também possui uma maneira de avaliação das restrições, dada pela equação (2.4), que é a mesma utilizada pelo SS-MVFCSMdd e o SS-CARD.

Note que essa estratégia também pode ser aplicada ao SS-CARD mas, para este trabalho, optamos por não alterar o funcionamento do SS-CARD afim de comparar a performance original do modelo com o modelo aqui proposto.

Algoritmo 13: SS-MVFCVSMdd Fuzzy Membership Update

for $i = 1$ to n **do**
  $A_i = \emptyset$
  $A_i = \left\{ k \in \{1, \ldots, C\} : \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_{kj}} d_j(e_i, e) = 0 \right\}$
  if $A_i \neq \emptyset$ **then**
    $u_{ik} = \frac{1}{|A_i|}, \forall k \in A_i$
    $u_{ir} = 0, \forall r \notin A_i$
  else
    $V_i = \{1, \ldots, C\}$
    for $k = 1$ to C **do**
      $a_{ik} = 2 \sum_{j=1}^{P} \lambda_{kj} \sum_{e \in G_{kj}} d_j(e_i, e);$
      $b_{ik} = 2\alpha \left(\sum_{(e_i, e_m) \in \mathcal{M}} \sum_{\substack{s=1 \\ s \neq k}}^{C} u_{ms} + \sum_{(e_i, e_m) \in \mathcal{C}} u_{mk} \right)$
    end for
    repeat
      $test = 0$
      $\gamma_i = \frac{1 + \sum_{w \in V_i} \frac{b_{iw}}{a_{iw}}}{\sum_{w \in V_i} \frac{1}{a_{iw}}};$
      for $k = 1$ to C **do**
        if $k \in V_i$ **then**
          $u_{ik} = \frac{\gamma_i - b_{ik}}{a_{ik}}$
          if $u_{ik} \leq 0$ **then**
            $u_{ik} = 0;$
            $V_i = V_i \backslash \{k\};$
            $test = 1;$
          end if
        end if
      end for
    until $test \neq 1$
  end if
end for

Algoritmo 14: SS-MVFCVSMdd Algorithm

INPUT
- $\mathcal{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_j, \ldots, \mathbf{D}_P\}$: the set of P dissimilarity $(N \times N)$ matrices;
- C: the number of cluster
- T: the maximum number of iterations;
- $m = 2$: mebership fuzziness parameter
- $0 < \epsilon << 1$: stopping parameter
- $q_j \, (j = 1, \ldots, P)$: cardinal of the set-medoids
- $\alpha > 0$: parameter related to the importance of the supervision (constraints)
- $\mathcal{M}$: the set of must-link constraints
- $\mathcal{C}$: the set of cannot-link constraints

OUTPUT
- the C-dimensional vector of vector of set-medoids $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_k, \ldots, \mathbf{G}_C)$
- the C-dimensional vector of relevance weight vectors $\mathbf{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_C)$
- the N-dimensional vector of membership degree vectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N)$

INITIALIZATION
- Set $t = 0$
- Randomly select C distinct vectors of set-medoids $\mathbf{G}_k \in E^{(q_1)} \times \ldots \times E^{(q_P)}$, $(1 \leq k \leq C)$
- Set $\boldsymbol{\lambda}_k = (1, \ldots, 1)$, $(1 \leq k \leq C)$
- Randomly set $u_{ik} \in [0, 1]$, with $\sum_{k=1}^{C} u_{ik} = 1$
- Compute J according to equation (3.5)

repeat
- Set $t = t + 1$
- **Representation step**
  - Compute the set-medoids G_{kj}, j-th component of the vector of set-medoids $\mathbf{G}_k$ according to algorithm 9
- **Weighting step**
  - Compute the relevance weight λ_{kj} of the dissimilarity matrix $\mathbf{D}_j$ into the fuzzy cluster k according to algorithm (10)
- **Allocation step**
  - Compute the membership degree u_{ik} of object e_i into the fuzzy cluster k according to algorithm 13
- Set $JOLD = J$
- Compute J according to equation (3.5)
- Set $JNEW = J$

until $|JNEW - JOLD| < \epsilon$ or $t > T$

4 EXPERIMENTOS

Este capítulo provê uma avaliação do algoritmo proposto SS-MVFCVSMdd comparando-o com o SS-CARD e o SS-MVFCSMdd através de aplicações com conjuntos de dados descritos por múltiplas matrizes de dissimilaridade.

4.1 Metodologia

Neste trabalho os rótulos de uma porcentagem dos objetos (10%, 30% e 50%) de um dado conjunto de dados foram utilizados para produzir as restrições par-a-par. Mais precisamente, o conjunto de restrições *must-link* é formado pelos pares de objetos com os mesmo rótulos, enquanto que o conjunto de restrições *cannot-link* é formado pelos pares de objetos com rótulos diferentes.

Primeiramente, para um determinado conjunto de dados, os rótulos de uma porcentagem de objetos escolhidos aleatoriamente foram levados em conta afim de produzir o conjunto de restrições *must-link* e *cannot-link*. Em seguida, assim como em (MELO; CARVALHO, 2013), foi feita uma busca entre 1 e 100 para estimar um valor apropriado para o parâmetro α (ver Subseção 2.4.1); note que essa busca não foi feita para o SS-CARD, pois seu algoritmo atualiza o valor de α a cada iteração como mostrado na equação (2.7).

Uma vez fixado o conjunto de restrições e o parâmetro α (lembrando que o α do SS-CARD não é fixado, dado seu algoritmo) cada algoritmo foi executado 30 vezes e a melhor solução, aquela com o menor valor final da função objetivo, foi selecionada. Para a melhor solução computou-se as métricas Partition Coefficient (BEZDEK, 1981) e Modified Partition Coefficient (DAVE, 1996) para avaliar a partição *fuzzy*, enquanto que para avaliar a partição *hard* computou-se as métricas Adjusted Rand Index (HUBERT; ARABIE, 1985) e F-measure (BREIMAN et al., 1984). Foi necessário realizar a defuzzificação da matriz de pertinência para viabilizar o cálculo da partição *hard*, tal processo de defuzzificação foi feito utilizando-se *first of maxima*.

Todo o processo descrito nos dois últimos parágrafos foi repetido 30 vezes (em cada repetição, para cada porcentagem, um conjunto diferente de restrições e, portanto, um possível valor diferente para α) e, para cada métrica, computou-se sua média e desvio padrão.

O caso em que não houve supervisão é indicado pela porcentagem de objetos rotulados igual à 0%. Nesse caso particular, cada algoritmo foi executado 30 vezes e a melhor solução, aquela com o menor valor final da função objetivo, foi escolhida.

Em um determinado conjunto de dados, para cada uma das métricas, temos como resultado uma tabela com o valor da média e desvio padrão da métrica em questão para cada um dos algoritmos para cada porcentagem (0%, 10%, 30% e 50%). Foram aplicados

então testes de hipótese não paramétricos afim de verificar se há diferença estatística significativa entre os algoritmos.

O valor do parâmetro q do SS-CARD (veja a seção 2.2) usado foi de 1.6. Em âmbos, SS-MVFCSMdd e SS-MVFCVSMdd, a cardinalidade do conjunto de medoids usada foi 3. Para todos esses algoritmos o parâmetro de ϵ e o número máximo de iterações foram fixados em, respectivamente, 10^{-9} e 150.

Para os algoritmos SS-MVFCSMdd e SS-MVFCVSMdd a inicialização da matriz de pertinência foi aleatória, assim como é feito no SS-CARD. Essa abordagem foi escolhida pois a inicialização da matriz de pertinência descrita nesses algoritmos fica dependente somente dos medoids escolhidos inicialmente de forma aleatória, o que não proporciona uma grande variabilidade da matriz de pertinência para diferentes execuções desses algoritmos.

4.2 Medidas de performance

Afim de comparar o desempenho dos algoritmos algumas medidas de performance (métricas) foram utilizadas. Tanto métricas para avaliar as partições *fuzzy* quanto *hard* foram utilizadas; as métricas que avaliam partição *hard* escolhidas foram Adjusted Rand Index (HUBERT; ARABIE, 1985) e F-measure (BREIMAN et al., 1984); já as métricas que avaliam partição *fuzzy* escolhidas foram Partition Cofficient (BEZDEK; EHRLICH; FULL, 1984) e Modified Partition Coefficient (DAVE, 1996).

Esta Subseção dá uma visão geral sobre cada uma dessas métricas explicando-as de forma sucinta.

4.2.1 Matriz de contingência

A matriz de contingência é usada para registrar observações independentes de duas ou mais variáveis aleatórias, normalmente qualitativas. Algumas métricas escolhidas para este trabalho fazem uso dessa matriz. Seja $P = \{P_1, \ldots, P_i, \ldots, Pm\}$ a partição a priori em m classes e $Q = \{Q_1, \ldots, Q_i, \ldots, Q_K\}$ a partição *hard* em K grupos dada por um algoritmo de agrupamento. Temos então a matriz de contingência (também chamada de matriz de confusão) apresentada na tabela 1.

4.2.2 Adjusted Rand Index

O Adjusted Rand Index (ou Corrected Rand Index) é dado por:

Tabela 1 – Confusion Matrix

classes	cluster					
	Q_1	$\dots$	Q_i	$\dots$	Q_K	$\sum$
P_1	n_{11}	$\dots$	n_{1j}	$\dots$	n_{1K}	$n_{1\bullet} = \sum_{j=1}^{K} n_{1j}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
P_i	n_{i1}	$\dots$	n_{ij}	$\dots$	n_{iK}	$n_{i\bullet} = \sum_{j=1}^{K} n_{ij}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
P_m	n_{m1}	$\dots$	n_{mj}	$\dots$	n_{mK}	$n_{m\bullet} = \sum_{j=1}^{K} n_{mj}$
$\sum$	$n_{\bullet 1} = \sum_{i=1}^{m} n_{i1}$	$\dots$	$n_{\bullet j} = \sum_{i=1}^{m} n_{ij}$	$\dots$	$n_{\bullet K} = \sum_{i=1}^{m} n_{iK}$	$N = \sum_{i=1}^{m}\sum_{j=1}^{K} n_{ij}$

$$\text{CR} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{K}\binom{n_{ij}}{2} - \binom{N}{2}^{-1} \times S}{\frac{1}{2}[\sum_{i=1}^{m}\binom{n_{i\bullet}}{2} + \sum_{j=1}^{K}\binom{n_{\bullet j}}{2}] - \binom{N}{2}^{-1} \times S} \tag{4.1}$$

$$S = \sum_{i=1}^{m}\binom{n_{i\bullet}}{2}\sum_{j=1}^{K}\binom{n_{\bullet j}}{2} \tag{4.2}$$

Onde $\binom{n}{2} = n(n-1)/2$ e n_{ij} representa o número de objetos que estão na classe P_i e no grupo Q_j; $n_{i\bullet}$ indica o número de objetos na classe P_i; $n_{\bullet j}$ indica o número de objetos no grupo Q_j; por fim N é o número total de objetos no conjunto de dados.

O índice *CR* avalia o grau de concordância (similaridade) entre uma partição a priori e a partição produzida por um algoritmo de agrupamento. Esse índice não é sensível ao número de classes nas partições ou à distribuição dos objetos nas partições. Ele toma valores no intervalo $[-1, 1]$ onde o valor 1 indica uma concordância perfeita entre as partições, enquanto que valores perto de 0 ou negativos indicam que a concordância entre as partições foi encontrada por acaso (HUBERT; ARABIE, 1985).

4.2.3 F-measure

O índice F-measure, assim como o CR, também é computado a partir da matriz de confusão; para este trabalho usamos o F_1-*score*. Primeiramente, a **matriz F-measure** é computada, que por sua vez é uma matriz mxK onde cada célula é a média harmônica da precisão (*precision*) e cobertura (*recall*) para uma classe P_i e um grupo Q_j. Cada célula da matriz F-measure pode ser computada de acordo com a seguinte equação:

$$\text{F-measure}(P_i, Q_j) = 2\,\frac{Precision(P_i, Q_j) \times Recall(P_i, Q_j)}{Precision(P_i, Q_j) + Recall(P_i, Q_j)} \tag{4.3}$$

onde

$$Precision(P_i, Q_j) = \frac{n_{ij}}{n_{\bullet j}} = \frac{n_{ij}}{\sum_{i=1}^{m} n_{ij}} \tag{4.4}$$

e

$$Recall(P_i, Q_j) = \frac{n_{ij}}{n_{i\bullet}} = \frac{n_{ij}}{\sum_{j=1}^{K} n_{ij}} \tag{4.5}$$

Levando em conta as equações (4.4) e (4.5), a equação (4.3) pode ser reescrita, após certa manipulação matemática, da seguinte forma:

$$\text{F-measure}(P_i, Q_j) = \frac{2n_{ij}}{\sum_{h=1}^{m} n_{hj} + \sum_{v=1}^{K} n_{iv}} \tag{4.6}$$

A partir da matriz F-measure, o índice F-measure pode ser computado usando a equação (4.7). Este índice toma valores no intervalo $[0, 1]$ no qual o valor 1 indica uma concordância perfeitas entre as partições.

$$\text{F-measure}(P, Q) = \frac{1}{N} \sum_{i=1}^{m} n_{i\bullet} \max_{1 \leq j \leq K} \text{F-measure}(P_i, Q_j) \tag{4.7}$$

4.2.4 Partition Coefficient

O coeficiente de partição (*Partition Coefficient*) (BEZDEK, 1981) é uma medida de validação de partição que se baseia na minimização do conteudo global da interseção *fuzzy* par-a-par de uma dada matriz de petinência $\mathbf{U}$, com N objetos e K grupos. O coeficiente da partição PC pode ser calculado da seguinte maneira:

$$\text{PC} = \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ik}^2 / N \tag{4.8}$$

Essa medida tem as seguintes propriedades:

$$\begin{gathered} (1/K) \leq \text{PC} \leq 1 \\ \text{PC} = 1 \Leftrightarrow \mathbf{U} \text{ is hard} \\ \text{PC} = 1/K \Leftrightarrow u_{ik} = 1/K \forall i, k \end{gathered} \tag{4.9}$$

Essas propriedades mostram que PC pode assumir valores no intervalo $[1/K, 1]$; PC assume o valor de 1 se U for uma partição *hard*. Em contrapartida, PC assume o valor mínimo de $1/K$ que ocorre quando a partição U não apresenta discriminação clara dos objetos. Essas propriedades podem ser provadas facilmente com o uso de multiplicadores de Lagrange .

As desvantagens do *Partition Coefficient* são sua tendência monotônica e falta de uma conexão direta com algumas propriedades dos dados em si.

4.2.5 Modified Partition Coefficient

O coeficiente de partição modificado (*Modified Partition Coefficient*) (DAVE, 1996) é uma medida de validação de partição baseada no *Partition Coefficient* 4.2.4. O *Partition Coefficient* é modificado a fim de se eliminar a dependência no número de grupos K, através da aplicação de uma transformação linear. Essa transformação faz com que o *Modified Partition Coefficient* assuma valores no intervalo $[0, 1]$ em contraste com o intervalo de $[1/K, 1]$ do *Partition Coefficient*.

Dada uma matriz de pertinência U, com N objetos e K grupos, o *Modified Partition Coefficient* pode ser calculado a partir da equação (4.10) onde PC é calculado utilizando-se a equação (4.8).

$$\mathrm{MPC} = 1 - \frac{K}{K-1}(1 - \mathrm{PC}) \tag{4.10}$$

4.3 Conjuntos de Dados

Para este trabalho foram usados alguns conjuntos de dados afim de avaliar o desempenho do SS-MVFCVSMdd em relação ao SS-CARD e o SS-MVFCSMdd. Os conjuntos utilizados bem como algumas de seus características estão resumidos na tabela 2. Cada conjunto será explicado com mais detalhes a seguir.

Tabela 2 – Summary of Data Sets

Data set	N (num. of objs.)	C (a priori classes)	T (views)
Phoneme	2000	4	3
Img. Segment.	2310	7	2
M. Features	2000	10	3
Reuters	1200	6	5
Corel	3400	34	7
AWA	2000	50	6

4.3.1 Phoneme

O conjunto de dados *phoneme* [1] consiste de 2000 objetos rotulados e 151 colunas. As primeiras 150 colunas correspondem às frequências, log-periodogramas discretizados, enquanto que a última coluna é o número do rótulo de 1 à 5 correspondendo, respectivamente, aos seguintes fonemas da língua inglesa: *sh*, *iy*, *dcl*, *aa*, and *ao*. Há 400 objetos para cada classe.

Para comparar as trajetórias temporais uma função de dissimilaridade "longitudinal transversal"proposta por D'Urso e Vichi foi utilizada (D'URSO; VICHI, 1998; D'URSO, 2000).

[1] http://www.math.univ-toulouse.fr/staph/npfda/npfda-datasets.html

Os autores propuseram uma dissimilaridade que é uma combinação da dissimilaridade transversal, que compara a posição instantânea (tendência) de cada par de trajetória, e duas dissimilaridades transversais, baseadas em conceitos de velocidade e aceleração de uma trajetória temporal. Temos então que o conjunto de dados terá três visões: posição, velocidade e aceleração. As matrizes de dissimilaridade foram geradas utilizando-se a distância euclidiana para cada matriz pertinente.

4.3.2 Image segmentation

No conjunto de dados *image segmentation* [2] os objetos foram retirados aleatoriamente de um conjunto de dados de 7 categorias de imagens ao ar livre. As imagens foram segmentadas a mão afim de criar uma classificação para cada pixel. Cada objeto é uma região $3x3$. As imagens segmentadas manualmente se enquadram em 7 classes, sendo os rótulos: *sky*, *cement*, *window*, *brick*, *grass*, *foliage* e *path*. Cada classe contém 330 objetos, formando um total de 2310 objetos. Cada objeto é descrito por 16 atributos de valores reais.

Para este trabalho, duas matrizes de dissimilaridade foram computadas a partir dos 16 atributos. A primeira matriz provê aspectos da forma (*shape*) das imagens, que corresponde aos atributos 4 à 9, sendo eles: *short-line-density-5*, *short-line-density-2*, *vedge-mean*, *vegde-sd*, *hedge-mean* e *hedge-sd*. A segunda matriz diz respeito à aspectos *rgb* das imagens, que são os últimos 7 atributos, sendo eles: *intensity-mean*, *rawred-mean*, *rawblue-mean*, *rawgreen-mean*, *exred-mean*, *exblue-mean*, *exgreen-mean*, *value-mean*, *saturation-mean* e *hue-mean*. A distância euclidiana foi usada em ambos os casos para gerar as matrizes de dissimilaridade. Os primeiros 3 atributos descrevem posição e não foram usados.

4.3.3 Multiple features

O conjunto de dados *multiple features* [3] consiste de caracteres de numerais escritos a mão de 0 à 9 extraídos de uma coleção de mapas utilitários Holandeses. 200 padrões (objetos) por classe, para um total de 2000 padrões, foram digitalizados em imagens binárias. Esses dígitos são representados em termos de 6 conjuntos de características: *fou*, 76 coeficientes de Fourier dos formatos dos caracteres; *fac*, 216 correlações de silhueta; *kar*, 64 coeficientes de Karhunen-Love; *pix*, 240 médias de pixel em janelas 2×3; *zer*, 47 momentos de Zernike; *mor*, 6 características morfológicas. Para cada conjunto de característica, uma matriz de dissimilaridade foi computada utilizando a distância euclidiana.

[2] https://archive.ics.uci.edu/ml/datasets/Image+Segmentation
[3] https://archive.ics.uci.edu/ml/datasets/Multiple+Features

4.3.4 Reuters

O conjunto de dados *reuters* (AMINI; USUNIER; GOUTTE, 2009) contém características de atributos de documentos escritos em cinco línguas diferentes (inglês, francês, alemão, espanhol e italiano) mas compartilhando o mesmo conjunto de categorias. Afim de explorar informação disponível de outras línguas, foi usado tradução de automática (*Machine Translation*) para produzir traduções de cada documento na coleção para todas as outras línguas antes de indexar. Para cada língua, temos então as características dos atributos de cada documento escritos em uma dada língua bem como as características dos atributos dos documentos traduzidos para aquela língua.

O conjunto possui de 12 à 30 mil documentos por língua e 11 à 34 mil documentos por classe. Os documentos estão rotulados em seis categorias: C15, CCAT, E21, ECAT, GCAT e M11. Neste trabalho, utilizamos apenas os documentos de língua originalmente inglesa, que são um total de 18.758 documentos, e suas respectivas traduções automáticas para outras línguas nos proporcionando 5 visões.

Devido à dimensionalidade do conjunto original, decidi realizar uma re-amostragem obtendo 1200 documentos com 200 documentos de cada categoria. Também foi aplicada uma redução de dimensionalidade utilizando LSA (*Latent Semantic Analysis*) (DEERWESTER et al., 1990) pois os vetores de atributos originais eram esparsos e de alta dimensionalidade; a dimensão dos vetores foi reduzida para 100. Feito isso, para cada visão (língua) gerou-se as matrizes de dissimilaridade utilizando a distância euclidiana para um total de 5 matrizes de dissimilaridade.

4.3.5 Corel

O conjunto de dados *corel* [4] utilizado é um subconjunto do COREL [5]. Ele contém 34 categorias, cada uma com 100 imagens, para um total de 3400 imagens. O critério de seleção utilizado foi se as imagens continham um objeto de primeiro plano saliente. O propósito é que as imagens desse conjunto podem ser apropriadamente processadas por sistemas de recuperação de imagens baseados em segmentação. As categorias selecionadas do COREL foram 290, 700, 750, 770, 840, 1040, 1050, 1070, 1080, 1090, 1100, 1120, 1340, 1350, 1680, 2680, 2890, 3260, 3510, 3540, 3910, 4150, 4470, 4580, 4990, 5210, 5350, 5530, 5810, 5910, 6440, 6550, 6610, 6840. As imagens também foram redimensionadas de forma que $max(width, height) = 384$ e $min(width, height) = 256$.

As imagens são descritas por um conjunto de 7 características: *ColorHsvHistogram64*, histograma de cores com 64 atributos; *ColorLuvMoment123*, *color moment* com 9 atributos; *ColorHsvCoherence64*, *color coherence* com 128 atributos; *CoarsnessVector* e *Directionality*, relacionados à características da textura de Tamura (TAMURA; MORI; YAMAWAKI,

[4] https://www.cs.virginia.edu/ xj3a/research/CBIR/Download.htm
[5] https://archive.ics.uci.edu/ml/datasets/Corel+Image+Features

1978) com 10 e 8 atributos respectivamente; *WaveletTwtTexture*, *wavelet texture* com 104 atributos; *MRSAR*, textura MASAR com 15 atributos.

Para este trabalho, em cada visão utilizou-se a distância euclidiana para gerar as matrizes de dissimilaridade, exceto para *ColorHsvHistogram64* em que se utilizou a interseção de histograma (SWAIN; BALLARD, 1991). Dados dois histogramas M e I com B *bins*, a interseção de histograma pode ser calculada da seguinte forma:

$$H(I, M) = \frac{\sum_{j=1}^{B} \min(I_j, M_j)}{\sum_{j=1}^{B} M_j}$$

Onde I_j é o valor do *bin* j pra um dado histograma I. O resultado fica no intervalo $[0, 1]$, onde 1 indica que os histogramas são idênticos. Ou seja, uma medida de **similaridade**. Uma outra forma de calcular a interseção de histograma é a seguinte, se:

$$\sum_{j=1}^{B} I_j = \sum_{j=1}^{B} M_j = T \tag{4.11}$$

Então interseção de histograma é equivalente ao uso da soma das diferenças absolutas ou a distância *city-block* (Manhattan):

$$1 - H(I, M) = \frac{1}{2T} \sum_{j=1}^{B} |I_j - M_j|$$

Nota-se que, nesse caso, o resultado ainda é no intervalo $[0, 1]$ mas 1 indica que os histogramas são completamente distintos. Ou seja, uma medida de **dissimilaridade**, o que é desejável para os algoritmos utilizados neste trabalho. Afim de satisfazer (4.11), o *ColorHsvHistogram64* de cada objeto foi normalizado da seguinte forma:

$$I_i = \frac{I_i}{\sum_{j=1}^{B} I_j}$$

o que assegura que $\sum_{j=1}^{B} I_j = 1$ para qualquer objeto.

4.3.6 Animals with Attributes

O conjunto de dados *Animals with Attributes* (AWA) [6] é um conjunto de 30475 imagens de 50 categorias de animais diferentes, com cada categoria contendo pelo menos 92 imagens. Foram extraídos das imagens 6 tipos diferentes de características: histogramas de cor RGB, SIFT, rgSIFT, PHOG, SURF, e histogramas de auto-similaridade locais (LAMPERT; NICKISCH; HARMELING, 2009).

Devido a dimensionalidade do conjunto, especialmente no tocante ao número de classes, para este trabalho, foi utilizado um subconjunto do AWA: 40 imagens de cada categoria foram escolhidas aleatoriamente, para um total de 2000 imagens. Para cada uma

[6] http://attributes.kyb.tuebingen.mpg.de/

das características foi gerada uma matriz de dissimilaridade: a distância *city-block* foi utilizada para os histogramas de cor RGB e PHOG, enquanto que a distância euclidiana foi utilizada para as demais características, dando um total de 6 matrizes de dissimilaridade (visões).

5 RESULTADOS

Este capítulo expõe os resultados dos experimentos realizados com os algoritmos SS-CARD, SS-MVFCSMdd e o SS-MVFCVSMdd em alguns conjuntos de dados multi-view, afim de comparar seus resultados e avaliar o desempenho tanto das partições *fuzzy* quando das partições *hard*.

5.1 Phoneme

A tabela 3 apresenta a performance dos algoritmos no conjunto de dados *phoneme*. Para ambas métricas que computam a qualidade das partições *fuzzy* e *hard*, o SS-MVFCVSMdd teve a melhor performance geral, seguido pelo SS-MVFCSMdd não muito atrás e, por último, o SS-CARD. Nota-se também que, sem supervisão, os algoritmos não conseguiram discriminar devidamente os objetos, atribuindo uma pertinência praticamente igual para cada objeto em cada partição.

O SS-CARD atribuiu aos objetos graus de pertinência *fuzzy* ruins, não discriminando claramente os objetos, como se pode observar pelas métricas *fuzzy* na tabela. O SS-MVFCVSMdd teve um desvio padrão de $0,0$ para todas as métricas assim como o SS-MVFCVSMdd, o que indica que o algoritmo proposto é tão robusto quanto o SS-MVFCSMdd. Também pode-se observar que o SS-MVFCVSMdd teve uma partição *fuzzy* com qualidade um pouco melhor que a do SS-MVFCSMdd e, ao defuzzificar as partições, o SS-MVFCVSMdd apresenta uma partição *hard* melhor que a do SS-MVFCSMdd.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 5% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que apenas na F-measure rejeitou-se a hipótese nula de que não há diferença estatística significativa entre os algoritmos, com um valor-p de 0.03877.

Aplicou-se então o teste post-hoc de Nemenyi para a F-measure com um nível de significância de 5% e observou-se que a hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd).

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 10% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que em todas as métricas, exceto o Adjusted Rand Index, rejeitou-se a hipótese nula de que não há diferença estatística significativa entre os algoritmos.

Aplicou-se então o teste post-hoc de Nemenyi para todas as métricas, exceto o Adjusted Rand Index, com um nível de significância de 10% e observou-se que a hipótese nula foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) em todos os casos.

Tabela 3 – Phoneme Dataset: Performance of the Algorithms

	Partition coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.2000	0.2257 (0.0140)	0.2000 (0.0000)	0.2000 (0.0000)
SS-MVFCSMdd	**0.2002**	0.2804 (0.0000)	0.4428 (0.0000)	0.6028 (0.0000)
SS-MVFCVSMdd	0.2000	**0.2819 (0.0002)**	**0.4435 (0.0000)**	**0.6032 (0.0000)**
	Modified Partition coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.0321 (0.0175)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	**0.0002**	0.1005 (0.0000)	0.3035 (0.0000)	0.5035 (0.0000)
SS-MVFCVSMdd	0.0000	**0.1024 (0.0002)**	**0.3044 (0.0000)**	**0.5039 (0.0000)**
	Adjusted Rand Index: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0013	0.4778 (0.0668)	0.0010 (0.0019)	0.0000 (0.0000)
SS-MVFCSMdd	**0.3365**	0.6420 (0.0000)	0.7433 (0.0000)	0.7999 (0.0000)
SS-MVFCVSMdd	0.0000	**0.7367 (0.0021)**	**0.8332 (0.0000)**	**0.8669 (0.0000)**
	F-measure: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.3323	0.6728 (0.0533)	0.3336 (0.0031)	0.3333 (0.0000)
SS-MVFCSMdd	**0.5958**	0.8088 (0.0000)	0.8723 (0.0000)	0.9086 (0.0000)
SS-MVFCVSMdd	0.3333	**0.8614 (0.0010)**	**0.9206 (0.0000)**	**0.9405 (0.0000)**

5.2 Image segmentation

A tabela 4 apresenta a performance dos algoritmos para o conjunto de dados *image segmentation*. No geral, o SS-MVFCSMdd e o SS-MVFCVSMdd tiveram uma performance similar, ambos melhores que o SS-CARD. A introdução da supervisão claramente melhorou o SS-MVFCSMdd e o SS-MVFCVSMdd, enquanto que no SS-CARD observou-se apenas uma pequena melhora.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 5% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que para todas as métricas a hipótese nula de que não há diferença estatística significativa entre os algoritmos foi rejeitada para todas as métricas.

Aplicou-se então o teste post-hoc de Nemenyi para todas as métricas com um nível de significância de 5%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) em todas as métricas, exceto o Adjusted Rand Index no qual a hipótese nula não foi rejeitada em nenhum dos pares.

Visto que, para o teste de hipótese de Friedman, a hipótese nula foi rejeitada para todas as métricas com um nível de significância de 5%, é de se esperar que a hipótese nula também seja rejeitada com um nível de significância de 10%.

Ao aplicar o teste post-hoc de Nemenyi para todas as métricas com um nível de

significância de 10%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada para o par (SS-CARD, SS-MVFCVSMdd) em todas as métricas, já para o par (SS-CARD, SS-MVFCSMdd) a hipótese nula foi rejeitada apenas para o Adjusted Rand Index.

Tabela 4 – Image Segmentation Dataset: Performance of the Algorithms

	Partition coefficient: mean and standard deviation			
Algorithm	0%	10%	30%	50%
SS-CARD	0.3045	0.1845 (0.0110)	0.3414 (0.0195)	0.3602 (0.0291)
SS-MVFCSMdd	0.2965	0.3655 (0.0005)	0.4922 (0.0000)	0.6372 (0.0000)
SS-MVFCVSMdd	**0.3171**	**0.3677 (0.0005)**	**0.4931 (0.0000)**	**0.6426 (0.0000)**
	Modified Partition coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.1885	0.0486 (0.0128)	0.2316 (0.0227)	0.2536 (0.0340)
SS-MVFCSMdd	0.1793	0.2597 (0.0005)	0.4075 (0.0000)	0.5767 (0.0000)
SS-MVFCVSMdd	**0.2032**	**0.2623 (0.0006)**	**0.4086 (0.0000)**	**0.5830 (0.0000)**
	Adjusted Rand Index: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0866	0.2333 (0.0656)	0.1707 (0.0207)	0.1733 (0.0218)
SS-MVFCSMdd	0.3654	**0.4977 (0.0037)**	0.5877 (0.0000)	**0.6876 (0.0000)**
SS-MVFCVSMdd	**0.4656**	0.4958 (0.0008)	**0.5920 (0.0004)**	0.6780 (0.0021)
	F-measure: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.3111	0.4536 (0.0510)	0.4178 (0.0243)	0.4089 (0.0279)
SS-MVFCSMdd	0.5686	0.6788 (0.0039)	0.7690 (0.0000)	**0.8421 (0.0000)**
SS-MVFCVSMdd	**0.6421**	**0.6831 (0.0017)**	**0.7739 (0.0004)**	0.8371 (0.0013)

5.3 Multiple features

A tabela 5 apresenta a performance dos algoritmos para o conjunto de dados *multiple features*. No geral, o SS-MVFCSMdd teve o melhor desempenho no tocante à partições *fuzzy* enquanto que o SS-MVFCVSMdd teve o melhor desempenho em relação à partições *hard*. Para os algoritmos supracitados, observa-se uma má performance quando não há supervisão (restrições) e melhora gradual a medida que mais restrições são adicionadas. Já o SS-CARD teve uma performance ruim, mesmo quando a supervisão estava presente, sendo incapaz de discriminar os objetos. O SS-MVFCSMdd e o SS-MVFCVSMdd apresentaram partições *hard* de melhor qualidade quando as restrições foram introduzidas.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 5% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que para todas as métricas a hipótese nula de que não há diferença estatística significativa entre os algoritmos foi rejeitada para todas as métricas.

Aplicou-se então o teste post-hoc de Nemenyi para todas as métricas com um nível de significância de 5%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) para as métricas F-measure e Adjusted Rand Index; enquanto que para as métricas Partition Coefficient e Modified Paritition Coefficient a hipótese nula foi rejeitada apenas para o par (SS-CARD, SS-MVFCSMdd).

Visto que, para o teste de hipótese de Friedman, a hipótese nula foi rejeitada para todas as métricas com um nível de significância de 5%, é de se esperar que a hipótese nula também seja rejeitada com um nível de significância de 10%.

Ao aplicar o teste post-hoc de Nemenyi para todas as métricas com um nível de significância de 10%, a rejeição da hipótese nula foi similar à observada com um nível de significância de 5%.

Tabela 5 – Multiple Features Dataset: Performance of the Algorithms

	Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.1000	0.1000 (0.0000)	0.1000 (0.0000)	0.1000 (0.0000)
SS-MVFCSMdd	**0.1006**	**0.2000 (0.0000)**	**0.3810 (0.0000)**	**0.5593 (0.0000)**
SS-MVFCVSMdd	0.1000	0.1994 (0.0000)	0.3809 (0.0000)	**0.5593 (0.0000)**
	Modified Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	**0.0007**	**0.1111 (0.0000)**	**0.3122 (0.0000)**	**0.5104 (0.0000)**
SS-MVFCVSMdd	0.0000	0.1105 (0.0000)	**0.3122 (0.0000)**	0.5103 (0.0000)
	Adjusted Rand Index: mean and standard deviation(in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.0007 (0.0002)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	0.0977	**0.8778 (0.0000)**	0.8616 (0.0000)	0.8863 (0.0000)
SS-MVFCVSMdd	**0.1741**	0.8673 (0.0000)	**0.8738 (0.0000)**	**0.8989 (0.0000)**
	F-measure: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.1818	0.1813 (0.0003)	0.1818 (0.0000)	0.1818 (0.0000)
SS-MVFCSMdd	0.2749	**0.9427 (0.0000)**	0.9349 (0.0000)	0.9466 (0.0000)
SS-MVFCVSMdd	**0.3350**	0.9375 (0.0000)	**0.9414 (0.0000)**	**0.9531 (0.0000)**

5.4 Reuters

A tabela 6 apresenta a performance dos algoritmos para o conjunto de dados *reuters*. Quando não há presença de restrições, os algoritmos têm uma performance similar e ruim; nenhum dos algoritmos foi capaz de discriminar os objetos. Ao introduzir restrições o SS-MVFCSMdd e o SS-MVFCVSMdd apresentam uma melhora bastante significativa

enquanto que o SS-CARD se mantém incapaz de discriminar os objetos não apresentando melhora.

O SS-MVFCSMdd e o SS-MVFCVSMdd tem performance similar, tanto para métricas *fuzzy* quanto *hard*. Vale ressaltar que observa-se uma qualidade significativamente melhor da partição *hard* no SS-MVFCVSMdd em relação ao SS-MVFCSMdd para 30% e 50% de objetos usados na geração de restrições.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 5% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que para todas as métricas a hipótese nula de que não há diferença estatística significativa entre os algoritmos não foi rejeitada. Tendo isso em vista, não foi necessário aplicar o teste post-hoc de Nemenyi, nesse caso.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 10% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que para todas as métricas a hipótese nula de que não há diferença estatística significativa entre os algoritmos foi rejeitada para todas as métricas. Contudo, ao aplicar o teste post-hoc de Nemenyi, a hipótese nula não foi rejeitada para nenhum dos pares em nenhuma das métricas.

Tabela 6 – Reuters Dataset: Performance of the Algorithms

	Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.1667	0.1671 (0.0001)	0.1667 (0.0000)	0.1667 (0.0000)
SS-MVFCSMdd	0.1667	0.2512 (0.0001)	**0.4177 (0.0000)**	**0.5842 (0.0000)**
SS-MVFCVSMdd	0.1667	**0.2516 (0.0000)**	**0.4177 (0.0000)**	0.5840 (0.0000)
	Modified Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.0006 (0.0001)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	0.0000	0.1014 (0.0001)	**0.3012 (0.0000)**	**0.5011 (0.0000)**
SS-MVFCVSMdd	0.0000	**0.1019 (0.0000)**	**0.3012 (0.0000)**	0.5008 (0.0000)
	Adjusted Rand Index: mean and standard deviation(in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.2946 (0.0437)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	0.0000	**0.3617 (0.0031)**	0.4780 (0.0000)	0.6097 (0.0000)
SS-MVFCVSMdd	0.0000	0.3434 (0.0000)	**0.5180 (0.0000)**	**0.6260 (0.0000)**
	F-measure: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.2857	0.5502 (0.0449)	0.2857 (0.0000)	0.2857 (0.0000)
SS-MVFCSMdd	0.2857	**0.6489 (0.0023)**	0.7409 (0.0000)	0.8197 (0.0000)
SS-MVFCVSMdd	0.2857	0.6386 (0.0000)	**0.7655 (0.0000)**	**0.8290 (0.0000)**

5.5 Corel

A tabela 7 apresenta a performance dos algoritmos para o conjunto de dados *corel*. Na ausência de supervisão, nenhum dos algoritmos conseguiu discriminar bem o objetos. Na presença de supervisão, podemos observar que o SS-CARD teve uma performance inferior aos outros dois algoritmos. O SS-MVFCSMdd e o SS-MVFCVSMdd tiveram uma performance parecida, mas observa-se que o último teve valores significativamente melhores para as métricas *hard*.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 5% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que apenas para as métricas F-measure e Adjusted Rand Index a hipótese nula de que não há diferença estatística significativa entre os algoritmos foi rejeitada.

Aplicou-se então o teste post-hoc de Nemenyi para a F-measure e o Adjusted Rand Index com um nível de significância de 5%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) para ambas as métricas.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 10% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que a hipótese nula foi rejeitada para todas as métricas.

Aplicou-se então o teste post-hoc de Nemenyi para a todas as métricas com um nível de significância de 10%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) para a F-measure e o Adjusted Rand Index, assim como observado para um nível de significância de 5%; já para as outras duas métricas a hipótese nula não foi rejeitada para nenhum dos pares de algoritmos.

Tabela 7 – Corel Dataset: Performance of the Algorithms

	Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0294	0.0294 (0.0000)	0.0294 (0.0000)	0.0294 (0.0000)
SS-MVFCSMdd	0.0294	0.1276 (0.0000)	**0.3216 (0.0000)**	**0.5155 (0.0000)**
SS-MVFCVSMdd	0.0294	**0.1277 (0.0000)**	**0.3216 (0.0000)**	**0.5155 (0.0000)**
	Modified Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.1011 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	0.0000	0.1011 (0.0000)	0.3010 (0.0000)	0.5008 (0.0000)
SS-MVFCVSMdd	0.0000	**0.1013 (0.0000)**	**0.3011 (0.0000)**	**0.5009 (0.0000)**
	Adjusted Rand Index: mean and standard deviation(in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0193	0.0210 (0.0019)	0.0213 (0.0017)	0.0208 (0.0018)
SS-MVFCSMdd	0.0000	0.2596 (0.0016)	0.3845 (0.0008)	0.5144 (0.0007)
SS-MVFCVSMdd	**0.0599**	**0.2812 (0.0002)**	**0.4077 (0.0002)**	**0.5384 (0.0000)**
	F-measure: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0845	0.0895 (0.0028)	0.0886 (0.0027)	0.0890 (0.0027)
SS-MVFCSMdd	0.0571	0.4449 (0.0021)	0.5977 (0.0005)	0.7128 (0.0003)
SS-MVFCVSMdd	**0.1544**	**0.4672 (0.0002)**	**0.6191 (0.0002)**	**0.7298 (0.0000)**

5.6 Animals with Attributes

A tabela 8 apresenta a performance dos algoritmos para o conjunto de dados *Animals with Attributes* (AWA). O SS-CARD não conseguiu discriminar os objetos, tendo a pior performance em relação aos outros algoritmos. O SS-MVFCSMdd e o SS-MVFCVSMdd tiveram performance, aparentemente, similar.

Para o Adjusted Rand Index, o SS-MVFCVSMdd teve valores um pouco melhores que o SS-MVFCSMdd na presença de supervisão, enquanto que para a F-measure o SS-MVFCSMdd teve valores um pouco melhores que o SS-MVFCVSMdd, exceto para 10%. Vale ressaltar que essas diferenças são relativamente pequenas em números absolutos.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 5% para cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que apenas para as métricas F-measure e Adjusted Rand Index a hipótese nula de que não há diferença estatística significativa entre os algoritmos foi rejeitada.

Aplicou-se então o teste post-hoc de Nemenyi para a F-measure e o Adjusted Rand Index com um nível de significância de 5%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) para a F-measure, enquanto que a hipótese nula foi rejeitada apenas para o par (SS-CARD, SS-MVFCSMdd) para o Adjusted Rand Index.

Ao aplicar o teste de hipótese de Friedman com um nível de significância de 10% para

cada uma das métricas afim de verificar se há diferença estatística significativa entre os algoritmos, observou-se que a hipótese nula foi rejeitada para todas as métricas.

Aplicou-se então o teste post-hoc de Nemenyi para todas as métricas com um nível de significância de 10%. A hipótese nula de que não há diferença estatisticamente significativa entre os algoritmos foi rejeitada apenas para o par (SS-CARD, SS-MVFCVSMdd) para a F-measure, enquanto que a hipótese nula foi rejeitada apenas para o par (SS-CARD, SS-MVFCSMdd) para o Adjusted Rand Index. A hipótese nula não foi rejeitada para nenhum dos pares nas demais métricas.

Tabela 8 – Animals with Attributes Dataset: Performance of the Algorithms

	Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0200	0.0200 (0.0000)	0.0200 (0.0000)	0.0200 (0.0000)
SS-MVFCSMdd	0.0200	0.0948 (0.0004)	**0.3141 (0.0000)**	**0.5100 (0.0000)**
SS-MVFCVSMdd	0.0200	**0.1046 (0.0000)**	**0.3141 (0.0000)**	**0.5100 (0.0000)**
	Modified Partition Coefficient: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	0.0000	0.0763 (0.0004)	**0.3001 (0.0000)**	**0.5000 (0.0000)**
SS-MVFCVSMdd	0.0000	**0.0863 (0.0000)**	**0.3001 (0.0000)**	**0.5000 (0.0000)**
	Adjusted Rand Index: mean and standard deviation(in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0000	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
SS-MVFCSMdd	**0.0021**	0.0089 (0.0012)	0.0882 (0.0004)	0.2563 (0.0008)
SS-MVFCVSMdd	0.0014	**0.0095 (0.0011)**	**0.1077 (0.0003)**	**0.2663 (0.0000)**
	F-measure: mean and standard deviation (in parenthesis)			
Algorithm	0%	10%	30%	50%
SS-CARD	0.0392	0.0795 (0.0054)	0.0392 (0.0000)	0.0392 (0.0000)
SS-MVFCSMdd	**0.0469**	0.1515 (0.0013)	**0.3742 (0.0005)**	**0.5535 (0.0005)**
SS-MVFCVSMdd	0.0461	**0.1743 (0.0018)**	0.3663 (0.0003)	0.5497 (0.0000)

5.7 Exemplo de saída

Esta seção mostra os dados de saída do algoritmo SS-MVFCVSMdd para o conjunto *multiple features* com 10% dos objetos usados para construção das restrições, afim de melhor ilustrar o funcionamento do algoritmo. A tabela 9 mostra os medoids finais de cada partição para cada visão enquanto que a tabela 10 mostra os pesos de relevância de cada partição para cada visão.

No conjunto *multiple features* os objetos estão agrupados por classe, temos então que os primeiros 200 objetos são de uma classe, por sua vez os seguintes 200 objetos são de outra classe e assim sucessivamente. Note que o primeiro objeto seria o 0. Isso nos permite analisar com mais facilidade a tabela 9. Considerando o caso de uma dada visão e uma

dada partição os medoids pertecem, de fato, à mesma classe (possuem o mesmo rótulo). Isso pode ser observado em quase todos casos, exceto, por exemplo, para a visão *zer* na partição Q_9, os objetos 354 e 267 pertencem à mesma classe enquanto que o objeto 1066 pertence à uma classe diferente.

Para a partição Q_{10} todos os medoids são do intervalo $[0, 199]$; para a partição Q_9 quase todos os medoids são do intervalo $[200, 399]$, exceto o objeto 1066 na visão *zer*; para a partição Q_7 todos os medoids são do intervalo $[400, 599]$; para a partição Q_2 todos os medoids são do intervalo $[600, 799]$; para a partição Q_6 quase todos os medoids são do intervalo $[800, 999]$, exceto para os objetos 1222 e 1919 na visão *fou* e o objeto 1161 na visão *zer*; para a partição Q_1 quase todos os medoids são do intervalo $[1000, 1199]$, exceto para o objeto 830 na visão *fou* e os objetos 489 e 572 na visão *mor*; para a partição Q_8 quase todos os medoids são do intervalo $[1200, 1399]$, exceto os medoids da visão *fou* e os objetos 1946 e 1994 na visão *mor*; para a partição Q_3 todos os medoids são do intervalo $[1400, 1599]$; para a partição Q_5 todos os medoids são do intervalo $[1600, 1799]$; para a partição Q_4 quase todos os medoids são do intervalo $[1800, 1999]$, exceto pelo objeto 1206 na visão *mor* e o objeto 1161 na visão *zer*. Pode-ser observar que os medoids escolhidos foram, em sua grande maioria, consistentes.

Tabela 9 – Medoids de cada partição para cada visão.

cluster	view					
	fac	*fou*	*kar*	*mor*	*pix*	*zer*
Q_1	1050, 1011, 1047	1195, 1107, 830	1198, 1090, 1174	1170, 489, 572	1198, 1035, 1039	1096, 1167, 1179
Q_2	685, 775, 600	752, 692, 622	669, 654, 763	771, 427, 748	664, 669, 654	668, 673, 693
Q_3	1592, 1412, 1433	1552, 1586, 1536	1471, 1535, 1481	1566, 1586, 1554	1440, 1582, 1481	1582, 1554, 1440
Q_4	1986, 1887, 1891	1890, 1847, 1831	1983, 1890, 1986	1206, 1815, 1868	1890, 1983, 1889	1902, 1161, 1889
Q_5	1798, 1792, 1740	1797, 1655, 1674	1615, 1743, 1792	1740, 1695, 1751	1792, 1659, 1683	1791, 1645, 1782
Q_6	901, 909, 840	1919, 926, 1222	909, 822, 923	884, 812, 942	909, 923, 838	1161, 914, 942
Q_7	460, 593, 585	587, 549, 435	460, 504, 493	481, 766, 509	504, 588, 462	560, 493, 436
Q_8	1227, 1347, 1226	1890, 1831, 1881	1238, 1292, 1227	1275, 1946, 1994	1238, 1381, 1227	1889, 1161, 1823
Q_9	361, 324, 221	296, 349, 374	254, 305, 363	326, 258, 1277	249, 351, 227	354, 267, 1066
Q_{10}	25, 85, 105	42, 197, 36	25, 80, 87	84, 194, 102	133, 80, 20	19, 76, 120

Na tabela 10 os pesos de relevância mostram que a visão *fou* tem relevância muito alta enquanto que as visões *fac* e *mor* possuem relevância acentuadamente baixa. Como descrito na seção 4.3.3, a visão *fou* diz respeito à 76 coeficientes de Fourier dos formatos dos caracteres e as visões *fac* e *mor* correspondem à 216 correlações de silhueta e 6 características morfológicas, respectivamente.

Tabela 10 – Pesos de relevância de cada partição para cada visão.

cluster	view					
	fac	*fou*	*kar*	*mor*	*pix*	*zer*
Q_1	0.108371	121.430588	4.203925	0.031202	2.188245	0.264747
Q_2	0.106927	125.858045	4.244331	0.030502	2.160703	0.265644
Q_3	0.084280	127.249166	3.967578	0.049604	2.026054	0.233844
Q_4	0.091202	118.610327	3.911342	0.046699	2.021539	0.250355
Q_5	0.102924	126.894755	3.563819	0.050256	1.874123	0.228106
Q_6	0.092720	119.275205	3.876268	0.046650	2.011101	0.248642
Q_7	0.107073	125.855543	3.876442	0.036182	2.012377	0.262916
Q_8	0.085802	118.284165	3.741129	0.055918	1.915549	0.245882
Q_9	0.089176	119.731556	3.908837	0.055205	2.009080	0.216032
Q_{10}	0.107692	132.890339	3.828815	0.042946	2.005172	0.211924

6 CONCLUSÃO

O presente trabalho introduziu o SS-MVFCVSMdd, um algoritmo semi-supervisionado *fuzzy c-medoids* para agrupamento *fuzzy* de dados relacionais *multi-view*, baseado no MVFCVSMdd mantendo características do mesmo e acrescentando restrições par-a-par *must-link* e *cannot-link* (semi-supervisão).

Assim como o MVFCVSMdd, o SS-MVFCVSMdd é capaz de inferir protótipos e pesos de relevância para cada visão além de também poder levar em conta restrições par-a-par que podem ser tanto fornecidas pelo usuário quanto extraídas a partir dos rótulos de pelo menos alguns objetos a serem considerados.

O SS-MVFCVSMdd se utiliza de multiplicadores de Lagrange e condições KKT na etapa de definição da melhor partição *fuzzy*, resultando numa solução algorítmica para esta etapa. Ao contrário do SS-CARD, o SS-MVFCVSMdd não precisa realizar *clipping* dos graus de pertinência de determinados objetos, pois as condições KKT asseguram que não haverá violação das restrições impostas nos graus de pertinência.

Foram realizados vários experimentos utilizando conjuntos de dados *multi-view* comparando o algoritmo proposto com algoritmos de características similares ao mesmo, foram eles: o SS-CARD e o SS-MVFCSMdd. Algumas métricas de qualidade de agrupamento, além de testes de hipótese sobre essas métricas, foram utilizadas e, ao observar os resultados, constatou-se que o algoritmo proposto teve performance similar ou superior em relação aos outros dois algoritmos.

Como trabalho futuro pode-se tentar aplicar restrições par-a-par às outras variações do MVFCVSMdd, tanto à de pesos locais quanto à de pesos globais; aplicar algum método para inicializar os pesos de relevância de forma aleatória; aplicar a busca de α (*self-learning*) utilizada pelo SS-MVFCSMdd e SS-MVFCVSMdd ao SS-CARD para fins de comparação; explorar mais o uso de *self-learning* no modelo proposto bem como o impacto de α no processo de agrupamento.

REFERÊNCIAS

AMINI, M.-R.; USUNIER, N.; GOUTTE, C. Learning from multiple partially observed views - an application to multilingual text categorization. In: *NIPS 22*. [S.l.: s.n.], 2009.

BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. 1. ed. Springer US, 1981. (Advanced Applications in Pattern Recognition). ISBN 978-1-4757-0452-5,978-1-4757-0450-1. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=98C46549B2E9025E9AD06D361F55D4C8>.

BEZDEK, J. C.; EHRLICH, R.; FULL, W. Fcm: Fuzzy c-means algorithm. *Computers and Geoscience*, Elsevier, v. 10, p. 191–203, 1984.

BICKEL, S.; SCHEFFER, T. Multi-view clustering. In: *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*. [S.l.: s.n.], 2004.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. *Classification and regression trees*. [S.l.]: CRC press, 1984.

CARVALHO, F. A. T. de; LECHEVALLIER, Y.; MELO, F. M. de. Relational partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices. *Fuzzy Sets and Systems*, Elsevier, v. 215, p. 1–28, 2013.

CARVALHO, F. A. T. de; MELO, F. M. de; LECHEVALLIER, Y. A multi-view relational fuzzy c-medoid vectors clustering algorithm. *Neurocomputing*, Elsevier, v. 163, p. 115–123, 2015.

CHAPELLE, O.; SCHOEKOPF, B.; ZIEN, A. *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

CLEUZIOU, G.; EXBRAYAT, M.; MARTIN, L.; SUBLEMONTIER, J.-H. Cofkm: A centralized method for multiple-view clustering. In: IEEE. *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. [S.l.], 2009. p. 752–757.

DAVE, R. N. Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recognition Letters*, Elsevier, v. 17, p. 613–623, 1996.

DEERWESTER, S.; DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American society for information science*, American Documentation Institute, v. 41, n. 6, p. 391, 1990.

D'URSO, P. Dissimilarity measures for time trajectories. *Journal of the Italian Statistical Society*, Springer, v. 9, n. 1-3, p. 53–83, 2000.

D'URSO, P.; VICHI, M. Dissimilarities between trajectories of a three-way longitudinal data set. In: *Advances in data science and classification*. [S.l.]: Springer, 1998. p. 585–592.

FRIGUI, H.; HWANG, C. Fuzzy clustering and aggregation of relational data with instance-level constraints. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 16, n. 6, p. 1565–1581, 2008.

FRIGUI, H.; HWANG, C.; RHEE, F. C.-H. Clustering and aggregation of relational data with applications to image database categorization. *Pattern Recognition*, Elsevier, v. 40, n. 11, p. 3053–3068, 2007.

GRIRA, N.; CRUCIANU, M.; BOUJEMAA, N. Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration. In: IEEE. *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05.* [S.l.], 2005. p. 867–872.

GRIRA, N.; CRUCIANU, M.; BOUJEMAA, N. Active semi-supervised fuzzy clustering. *Pattern Recognition*, Elsevier, v. 41, p. 1834–1844, 2008.

HATHAWAY, R. J.; BEZDEK, J. C. Nerf c-means: Non-euclidean relational fuzzy clustering. *Pattern recognition*, Elsevier, v. 27, n. 3, p. 429–437, 1994.

HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, Springer, v. 2, n. 1, p. 193–218, 1985.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, v. 31, p. 651–666, 2010.

JAIN, A. K.; MURTY, M.; FLYNN, P. Data clustering: a review. *ACM Computing Surveys*, v. 31, p. 233–264, 1999.

LAMPERT, C. H.; NICKISCH, H.; HARMELING, S. Learning to detect unseen object classes by between-class attribute transfer. In: IEEE. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* [S.l.], 2009. p. 951–958.

MELO, F. M. de; CARVALHO, F. A. T. de. Semi-supervised fuzzy c-medoids clustering algorithm with multiple prototype representation. In: IEEE. *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on.* [S.l.], 2013. p. 1–7.

ROUSSEEUW, P. J.; KAUFMAN, L. *Finding Groups in Data.* [S.l.]: Wiley Online Library, 1990.

SHENL, R.; OLSHEN, A. B.; LADANYI, M. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, v. 25, p. 2906–2912, 2009.

SWAIN, M. J.; BALLARD, D. H. Color indexing. *International journal of computer vision*, Springer, v. 7, n. 1, p. 11–32, 1991.

TAMURA, H.; MORI, S.; YAMAWAKI, T. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, v. 8, n. 6, p. 460–473, 1978.

TZORTZIS, G. F.; LIKAS, A. C. Multiple view clustering using a weighted combination of exemplar-based mixture models. *IEEE Transactions on Neural Networks*, v. 21, p. 1925–1938, 2010.

XU, R.; WUNUSCH, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, v. 16, p. 645–678, 2005.

APÊNDICE A – DERIVAÇÃO DAS EQUAÇÕES DE ATUALIZAÇÃO DA PARTIÇÃO FUZZY

O SS-MVFCVSMdd minimiza:

$$J = \sum_{k=1}^{C}\sum_{i=1}^{N}(u_{ik})^2 \sum_{j=1}^{P}\lambda_{kj}\sum_{e\in G_{kj}} d_j(e_i,e) + \alpha\left(\sum_{(l,m)\in\mathcal{M}}\sum_{r=1}^{C}\sum_{\substack{s=1\\s\neq r}}^{C} u_{lr}u_{ms} + \sum_{(l,m)\in\mathcal{C}}\sum_{r=1}^{C} u_{lr}u_{mr}\right) \tag{A.1}$$

Sujeito à:

$$u_{ik} \geq 0 \quad \sum_{k=1}^{C} = 1 \quad \forall i \quad \text{e} \quad \lambda_{kj} > 0 \quad \prod_{j=1}^{P}\lambda_{kj} = 1 \quad \forall k$$

Para minimizar a função objetivo J com respeito à matriz de pertinência U usamos o método dos multiplicadores de Lagrange, temos assim:

$$L = J - \sum_{i=1}^{N}\gamma_i\left(\sum_{k=1}^{C} u_{ik} - 1\right) - \sum_{i=1}^{N}\sum_{k=1}^{C}\psi_{ik}u_{ik} \tag{A.2}$$

onde γ_i e ψ_{ik} são os multiplicadores.

Calculando sua derivada em relação ao grau de pertinência encontramos:

$$\frac{\partial L}{\partial u_{ik}} = u_{ik}a_{ik} + b_{ik} - \psi_i - \gamma_{ik} \tag{A.3}$$

na qual

$$\begin{aligned} a_{ik} =& 2\sum_{j=1}^{P}\lambda_{kj}\sum_{e\in G_{kj}} d_j(e_i,e) \\ b_{ik} =& 2\alpha\left(\sum_{(i,m)\in\mathcal{M}}\sum_{r=1}^{C}\sum_{\substack{s=1\\s\neq r}}^{C} u_{ir}u_{ms} + \sum_{(l,i)\in\mathcal{C}}\sum_{r=1}^{C} u_{lr}u_{ir}\right) \end{aligned} \tag{A.4}$$

Os requisitos pra minimização da função (A.3) de acordo com as condições de Kuhn e Tucker são:

$$\begin{cases} \psi_{ik} \geq 0 \\ \frac{\partial L}{\partial uik} = 0 \\ u_{ik}\psi_{ik} = 0 \end{cases} \tag{A.5}$$

Resolvendo a relação (A.5) com $\sum_{k=1}^{C} u_{ik} = 1$ temos:

$$\gamma_i = \frac{1 + \sum_{w=1}^{C}(b_{ik}/a_{iw}) - \sum_{w=1}^{C}(\psi_{ik}/a_{iw})}{\sum_{w=1}^{C}(1/a_{iw})} \tag{A.6}$$

e

$$u_{ik} = \frac{\gamma_i + \psi_{ik} - b_{ik}}{a_{ik}} \tag{A.7}$$

Substituindo o termo (A.6) em (A.7), temos:

$$u_{ik} = \frac{(1/a_{ik})}{\sum_{k=1}^{C}(1/a_{iw})} + \frac{\sum_{k=1}^{C}(b_{iw}/a_{iw})}{a_{ik}\sum_{k=1}^{C}(1/a_{iw})} - \frac{b_{ik}}{a_{ik}} + \frac{\psi_{ik}}{a_{ik}} - \frac{\sum_{k=1}^{C}(\psi_{iw}/a_{iw})}{a_{ik}\sum_{k=1}^{C}(1/a_{iw})} \tag{A.8}$$

As condições (A.5) permitem a existência de apenas duas possibilidades:

$$\psi_{ik} = 0 \implies u_{ik} \geq 0 \tag{A.9}$$

$$\psi_{ik} > 0 \implies u_{ik} = 0 \tag{A.10}$$

Em (A.8) se considerarmos $\psi = 0$ e obtivermos $u_{ik} \leq 0$ podemos afirmar com segurança que apenas a possibilidade (A.10) pode ser satisfeita para que u_{ik} não seja negativo.

É possível que $a_{ik} = 0$, nesse caso particular, optamos por maximizar o u_{ik}, pois esse caso indica que a dissimilariade do objeto e_i em relação ao grupo k é zero. Nesse âmbito, temos o caso da equação (A.11) e resolvendo γ_i na equação (A.3) temos $\gamma_i = b_{ik}$. A solução em que existem diferentes valores de b_{ik} quando $a_{ik} = 0$ não é valida, então temos que fazer a escolha de algum b_{ik} para γ_i. Como ainda temos que manter a equação (A.5) como solução, isto é $\psi_{ik} \geq 0$, logo a única escolha de γ_i que torna isso possível é quando $\gamma_i = min(\{b_{ik} \,|\, a_{ik} = 0\})$. Com o valor de γ_i e usando a equação (A.7) temos:

$$\begin{cases} 0 \quad \text{se} \quad \gamma_i \leq b_{ik} \\ \frac{\gamma_i - b_{ik}}{a_{ik}} \quad \text{se} \quad a_{ik} > 0 \\ \frac{1 - \sum_{w \notin Z_i} u_{iw}}{|Z_i|} \quad \text{se} \quad a_{ik} = 0 \end{cases} \tag{A.11}$$

onde

$$Z_i = \{k \,|\, a_{ik} = 0\}$$

Note que na equação (A.11) a divisão foi igualitária para os elementos de Z_i mas outras escolhas também são possíveis. Se ainda existir algum elemento $u_{ik} < 0$ isso quer dizer que o valor de γ_i não foi válido ao mesmo tempo que qualquer outro valor maior ou igual ao que foi escolhido causará o mesmo problema logo temos que $\gamma_i < min(\{b_{ik} \mid a_{ik} = 0\})$. Mesmo que esse valor não seja exato essa condição faz com que se $a_{ik} = 0$ então $u_{ik} = 0$ através da equação (A.11), anulando o caso particular.

APÊNDICE B – ARTIGO PUBLICADO NA CONFERÊNCIA FUZZ-IEEE

Giovanni Acampora fuzzieee2017@ieee-cis.org por ditthi.pair.com
para mim, fatc

Dear Author(s),

Congratulations! On behalf of the FUZZ-IEEE 2017 Technical Program Committee and Technical Chairs, we are pleased to inform you that your paper:

Paper ID: 225
Author(s): Diogo Branco and Francisco de Carvalho
Title: Fuzzy Clustering of Multi-View Relational Data with Pairwise Constraints

has been accepted for presentation at the FUZZ-IEEE 2017 and for publication in the conference proceedings published by IEEE. This email provides you with all the information you require to complete your paper and submit it for inclusion in the proceedings. A notification of the presentation format (oral or poster) will be sent soon.

Figura 2 – Carta de aceitação do artigo na IEEE International Conference on Fuzzy Systems 2017

Fuzzy Clustering of Multi-View Relational Data with Pairwise Constraints

Diogo P. P. Branco
Centro de Informtica (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife, Pernambuco 50740-560
Email: dppb@cin.ufpe.br

Francisco de A. T. de Carvalho
Centro de Informtica (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife, Pernambuco 50740-560
Email: fatc@cin.ufpe.br

***Abstract*—This paper presents SS-MVFCVSMdd, a semi-supervised multiview fuzzy clustering algorithm for relational data described by multiple dissimilarity matrices. SS-MVFCVSMdd provides a fuzzy partition in a predetermined number of fuzzy clusters, a representative for each fuzzy cluster, learns a relevance weight for each dissimilarity matrix, and takes into account pairwise constraints *must-link* and *cannot-link*, by optimizing a suitable objective function. Experiments with multi-view real-valued data sets described by multiple dissimilarity matrices show the usefulness of the proposed algorithm.**

I. Introduction

Clustering is an essential machine learning task and it is widely applied in many fields such as pattern recognition, data mining, computer vision and bioinformatics. It aims to organize a set of objects into clusters such that objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible [1].

Clustering techniques often are divided into hierarchical and partitioning methods. Hierarchical methods provides a hierarchical structure of groups, i.e., a nested sequence of partitions of the input data often represented by a dendogram. Partitioning methods aim to provide a single partition of the input data in a fixed number of clusters, typically by optimizing an objective function measuring the total heterogeneity within cluster. They are performed mainly into strict partitioning (hard) and non-strict (soft or fuzzy) partitioning clustering. In hard clustering, any object may belong to one and only one cluster. On the other hand, in fuzzy clustering the objects are assigned to all clusters with a certain fuzzy membership degree [2], [3].

Conventional partitioning clustering algorithms often operates on a single data matrix where objects in rows are described by variables in columns (vector data). Although these methods have been deeply studied and shown to be very useful in practice, there is an increasing need to methods that are able to perform on objects described from different views, often involving data extracted from different sources with different sets of measurements and scales. For example, in tumor studies one may need to take into account simultaneously genomic, epigenomic, transcriptomic and proteomic data [4].

Currently, there are mainly three approaches to manage multi-view data in the clustering analysis. The first achieves concatenation or data fusion, in which the views can be either concatenated or merged and compiled in a single data table. The second, known as distributed approach, groups the views independently, using the same or different algorithms, and provides a consensus partition from the partition obtained with each individual view. Finally, a centralized approach, that takes into account all views simultaneously to provide a single partition of the data [5].

Several centralized multiview clustering techniques have been successfully applied on vector data [6], [7]. However, challenges still remains for these techniques. Indeed, the views may not be easily described as vector data, for example when the samples are video and audio from a camera or text and links of a web page, and thus they may be not directly comparable to one another. Moreover, even when the views could be expressed as vector data, there are still difficulties when the features have very different statistical properties.

Centralized multiview clustering techniques that operates on relational data described by multiple dissimilarity matrices [8]–[10] deal with these difficulties more easily, they need just a suitable dissimilarity measure aiming to describe the relationships between the samples according to each view. Relational data can be very useful when a particular dissimilarity measures are needed to the problem at hand, when confidentiality is needed, or when the nature of the data different.

Clustering is a difficult task, specially for large, high dimensional and multi sources data sets. Partial supervision, that often concerns either the membership of some objects to specific clusters, or pairwise constraints (must-link, cannot-link) between objects, can alleviate this problem [11], [12]. Refs. [13], [14] provide centralized multiview clustering techniques that operates on relational data described by multiple dissimilarity matrices with partial supervision.

This paper presents SS-MVFCVSMdd, a semi-supervised multiview fuzzy clustering algorithm for relational data described by multiple dissimilarity matrices. SS-MVFCVSMdd provides a fuzzy partition in a predetermined number of fuzzy clusters, a representative for each fuzzy cluster, learns a relevance weight for each dissimilarity matrix, and takes into account pairwise constraints *must-link* and *cannot-link*, by optimizing a suitable objective function. It extends Ref. [10] in

Figura 3 – Primeira página do artigo publicado na IEEE International Conference on Fuzzy Systems 2017