Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Metodologia para Agrupamento de Dados - Uma versão da Busca Tabu para Sistema R

Marcelus Santana Távora

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Brasília
2015

MARCELUS SANTANA TÁVORA

Metodologia para Agrupamento de Dados - Uma versão da Busca Tabu para Sistema R

Trabalho de Conclusão de Curso apresentado ao Departamento de Estatística da Universidade de Brasília, como parte dos requisitos para a obtenção do título de Bacharel em Estatística.

Orientadora: Profa. Dra **Maria Amelia Biagio**

Brasília
2015

Dedicatória

Dedico este trabalho primeiro a Deus. Também dedico à minha família, meus pais, meus irmãos e também a meus amigos.

Agradecimentos

Agradeço primeiramente a Deus por ter me ajudado a fazer este trabalho.

Agradeço à minha família pela paciência e compreensão das muitas horas de estudo que tive, deixando de participar de certos momentos com a mesma. Agradeço a mesma por sempre ter me incentivado a manter o foco no curso de Estatística.

Agradeço aos meus amigos do curso por terem tido a paciência em me ensinar e pelo companheirismo. Também agradeço pelos amigos que não fazem parte do curso, mas que sempre me deram forças.

Agradeço aos professores e funcionários do Departamento de Estatística da Universidade de Brasília por terem me ensinado bastante e pelo respeito que tiveram comigo e pela amizade.

Agradeço à Universidade de Brasília pela grande oportunidade que me foi dada de ter estudado numa universidade de ponta.

Resumo

Metodologias não-hierárquicas para agrupamento de dados têm sido bastante estudadas e utilizadas nas últimas décadas, muitas delas buscando otimizar um critério comum que é o de minimizar a soma dos quadrados das discrepâncias internas aos grupos formados por seus procedimentos. O problema de se agrupar dados com este critério é bastante conhecido na literatura como problema MSSC (Minimum Sum of Squares Clustering). Dentre as metodologias voltadas para a resolução de problema MSSC deve-se citar a já bastante conhecida heurística K-Means. Com o mesmo propósito, muitas metodologias surgiram nas últimas décadas e, dentre as principais, destacam-se as metodologias H-Means, e mais recentemente sua forma não-degenerada H-Means+, as metodologias Tabu Search e VNS. No entanto, em ambiente computacional fortemente demandado por estatísticos, como o sistema computacional R, estas metodologias, com exceção da primeira, ainda não estão disponíveis. O presente trabalho consiste no desenvolvimento computacional, através da linguagem do sistema em referência (R Core Team, 2014), de uma versão híbrida da metodologia de agrupamento não-hierárquico Busca Tabu (ou Tabu Search) com a heurística H-Means+, esta última recentemente implementada em linguagem do sistema R. Resultados computacionais são obtidos para os bancos de dados USArrests e Íris de Fisher, ambos disponíveis no mesmo sistema em referência. Análise comparativa dos agrupamentos, obtidos pelas metodologias K-Means, H-Means+, a versão implementada e denominada HBaseTabu, e versão híbrida das duas primeiras, denominada HK-Means, é apresentada para distintos números de clusters. Os resultados apresentados são validados, no primeiro teste, através dos valores ótimos apresentados por K-Means, e por valores ótimos já conhecidos para os testes realizados com o banco de dados Iris. Através da análise dos resultados obtidos, pode-se observar que a heurística proposta e implementada neste trabalho apresenta resultados compatíveis com aqueles obtidos por K-Means e demonstra, em vários casos, superioridade sobre as heurísticas H-Means+ e HK-Means. Em situações em que o número de agrupamentos é maior que três. Em alguns casos os resultados obtidos mostraram-se melhores que os apresentados pela heurística K-Means,

quando inicializou-se a metodologia implementada, HBaseTabu, com a melhor solução apresentada por K-Means, o que demonstra o poder de eficiência computacional da proposta HBaseTabu.

Palavras-chave: Agrupamento de Dados, problemas MSSC, heurística Busca Tabu, linguagem de computação S

Sumário

Lista de Tabelas

Introdução

Análise de Agrupamento de Dados (ou Cluster Analysis) é uma área da Estatística voltada para o particionamento de conjuntos de dados, tendo como principal objetivo facilitar a leitura e compreensão dos padrões das características apresentadas pelos elementos estudados. Portanto, é importante enfatizar que seu objetivo antecede a classificação dos dados em estudo, visto que requer, em primeiro lugar, a identificação das classes ou grupos existentes no conjunto de dados. Particularmente, seu objetivo é designar os elementos de um conjunto de dados em grupos mutuamente excludentes, observando suas similaridades (ou dissimilaridades).

O problema de se determinar agrupamento de dados é bastante conhecido e utilizado em muitas áreas do saber, tais como matemática, ciências da computação, astronomia, geologia, pesquisa de mercado, medicina, etc.

Como exemplos, pode-se citar: na área da Psiquiatria, o agrupamento de dados auxilia técnicas para refinar ou até mesmo para redefinir tipos de diagnósticos atuais. Pilowksy et al (1969), utiliza um método apresentado em Wallace e Boulton (1968), em 200 pacientes deprimidos a fim de descobrir a existência de subtipos deprimidos, tais como os endógenos e os neuróticos; na Astronomia, os profissionais da área necessitam determinar o numero e o tipo de classes diferentes de estrelas existentes. O conseqüente exercício de classificação de estrelas pode também ajudar na identificação de objetos desconhecidos no espaço dentro de um número gigante de dados. Como outro exemplo, na área de Pesquisa de Mercado (Chakrapani, 2004), pode-se citar o caso de uma fabricante de automóveis que acreditava que a compra de um carro esportivo era influenciada pelo estilo de vida do cliente e não pela sua idade ou outros motivos. Então, a fabricante utilizou a área de agrupamento de dados para identificar clientes com o estilo de vida ligado à compra de modelos esportivos.

Por ser de natureza altamente combinatorial, o problema de se determinar agrupamentos em conjunto de dados teve grande avanço após o aparecimento do computador e, mais recentemente, dos computadores de nova geração que possuem grande capacidade e agilidade em efetuar milhares de operações matemáticas. Nas últimas

décadas, muitas foram as metodologias desenvolvidas, as quais apresentaram diferentes maneiras e critérios para a determinação do melhor (ou mais adequado) agrupamento de um conjunto de dados. Dentre as muitas metodologias existentes, encontram-se aquelas que buscam obter o agrupamento atraves da resolução do problema MSSC (Minimum Sum of Squares for Clustering), comumente utilizado na literatura. Sua resolução não é trivial, dado que o mesmo possui muitos mínimos locais, e este fato tem estimulado estudiosos do assunto a proporem metodologias que sejam capazes de apresentar soluções próximas da solução ótima global.

Dentre as metodologias mais conhecidas, e que apresentam solução para o problema MSSC destacam-se as heurísticas K-Means (McQueen, [3]) e H-Means (Forgy, [8]) e, mais recentemente, a heurística Busca Tabu (Al Sultan, 1995), que possue como um dos propósitos alcançar soluções que não sejam apenas soluções locais.

O presente trabalho tem como objetivo estudar algumas metodologias voltadas para a resolução do problema MSSC, formular e implementar, no sistema de computação estatística R, uma heurística que se baseia em princípios da Busca Tabu, denominada BaseTabu, e que explore, ao mesmo tempo, a eficiência operacional das heurísticas de McQueen e Forgy acima citadas, incluindo heurísticas híbridas dessas duas últimas, como a heurística HK-Means e a heurística HK-BaseTabu, que é aquela proposta e inicializada por HK-Means. Com esta finalidade, este trabalho é organizado da seguinte maneira: o capitulo 1 apresenta aspectos do Estado da Arte, o capitulo 2 as metodologias estudadas e aquela formulada e implementada, o capitulo 3 resultados computacionais obtidos e análise comparativa entre todos os métodos do estudo e o capitulo 4 conclusões acerca dos resultados obtidos e as considerações finais do trabalho.

Objetivo Geral

O objetivo deste trabalho é formular uma heurística híbrida, que seja uma junção da metodologia Busca Tabu (ou Tabu Search) e a heurística H-Means, todas voltadas para a resolução do problema MSSC (Minimum Sum of Squares for Clustering), em linguagem compatível à do sistema de operação estatística R.

Objetivos específicos

Comparar a nova metodologia com outras metodologias clássicas de agrupamento de dados, disponíveis no mesmo sistema.

Além do aprendizado do aluno sobre algumas das principais metodologias para agrupamento de dados e da linguagem do sistema de computação estatística R,

espera-se obter boa adaptação da heurística proposta ao ambiente estatístico R e que as soluções indicadas por esta versão sejam tão boas ou melhores quanto aquelas indicadas pelos métodos não-hierárquicos K-Means, disponível no mesmo sistema computacional, e H-Means, este último recentemente programado em linguagem R [Felipe Quintino, PIBIC 2014/2015].

Capítulo 1

Aspectos do Estado da Arte

Dada a natureza combinatorial do problema de agrupamento de dados, o boom da tecnologia da informática estimulou enormemente o desenvolvimento de estudos e pesquisas relacionados à resolução deste problema. Consequentemente, nas últimas décadas, muitos foram os estudiosos que propuseram metodologias para a resolução do problema em referência; dentre eles deve-se citar alguns pioneiros no assunto como Florek et al. (1951), Johnson (1967), Lance e Williams (1967), e Ward (1963) (vide Kaufman e Rousseeuw (1990)). Levantamentos bibliográficos sobre as várias metodologias existentes podem ser encontrados em Aloise e Hansen, 2008 [1] e Jain, 2010 [11].

Todas as metodologias para agrupamento de dados necessitam utilizar medidas de similaridade (ou dissimilaridade) entre os elementos estudados e um critério para obtenção dos grupos, os quais devem ser escolhidos de acordo com a natureza (se qualitativos e/ou quantitativos) e disposição dos elementos no conjunto de dados. Elas diferem bastante entre si e podem ser divididas, segundo a literatura, em dois grandes grupos: as Hierárquicas e as Não-hierárquicas.

As metodologias Hierárquicas necessitam a princípio da informação sobre a matriz de similaridade (ou dissimilaridade) entre os elementos observados e pertencentes ao conjunto de dados. Elas são classificadas em Aglomerativas e Divisivas. As Aglomerativas tomam, inicialmente, tantos grupos quantos forem os elementos a serem estudados e, a cada passo, de acordo com um critério, une aqueles mais similares para formarem um novo grupo, e assim segue o processo até obter apenas um grupo com todos os indivíduos do conjunto de dados. De acordo com o critério de aglomeração utilizado, as metodologias Hierárquicas Aglomerativas recebem as seguintes denominações: Ligação do Vizinho mais Próximo, Ligação do Vizinho mais Longe, Media dos Grupos e a de Ward [12].

Com procedimento inverso ao utilizado pelas últimas, as Hierárquicas Divisivas inicializam o procedimento com um grupo formado por todos os elementos em

estudo e, passo a passo, procede na divisão do grupo mais dissimilar em outros dois grupos a fim de obter, no final, tantos grupos quanto forem os elementos do conjunto de dados. Este processo de divisão dos grupos demanda grande carga computacional; por esta razão, as metodologias Aglomerativas tem sido preferidas por apresentarem maior simplicidade e eficiência de implementação.

As metodologias Não-hierárquicas não necessitam da informação da matriz de similaridade para iniciarem o procedimento mas requerem, como parâmetro, o número de agrupamentos, e buscam encontrar a melhor partição, para aquele número, existente no conjunto de dados. Nestas, iterativamente, as partições são obtidas de maneira a otimizar um determinado critério, que possui como objetivo alcançar maior homogeneidade dentro grupos (ou maior heterogeneidade entre os grupos). Dessa forma, passo a passo, elementos pertencentes a um grupo podem ser designados a outros grupos desde que o critério supra citado seja satisfeito. Assim, por apresentarem maior flexibilidade computacional que as Hierárquicas, as metodologias Não-Hierárquicas vem sendo exaustivamente estudadas com o objetivo de se tentar alcançar melhores soluções para o problema de agrupamento de dados.

Um dos objetivos comumente utilizado pelas metodologias Não- Hierárquicas é a minimização da soma dos quadrados das discrepâncias dentro dos grupos; com isso, o problema de encontrar a melhor partição em um conjunto de dados passa a ser denominado, na literatura, como um problema MSSC (Minimum Sum of Squares Clustering). Sua resolução não é trivial, dado que o mesmo possui muitos mínimos locais, e este fato tem estimulado estudiosos do assunto a proporem metodologias que, associadas ou não às Não-hierárquicas, sejam capazes de apresentar soluções próximas da solução ótima global.

Dentre as metodologias mais conhecidas, voltadas para a resolução de problemas MSSC, estão as heurísticas K-Means (McQueen, 1967) e H-Means (Forgy, 1965). O fato deste problema ser de resolução complexa, apresentando muitos mínimos locais, muitas outras metodologias surgiram com o propósito de alcançar soluções melhores, procurando explorar vizinhanças distintas daquela onde um mínimo local é geralmente encontrado. Dentre elas, pode-se citar as heurísticas VNS (Hansen e Mladenovic, 2001) e Busca Tabu (Al Sultan, 1995, Glover e Laguna, 2002) (ambas amplamente utilizadas em problemas de distintas áreas do saber [2,8,9,10]), que possuem como ponto comum a exploração de vizinhanças cada vez mais distantes daquela onde se encontra um mínimo local conhecido. Para alcançar regiões distintas do conjunto solução do problema em questão, estas últimas constroem novas partições do conjunto de dados a partir de perturbações da partição correspondente a uma determinada solução, a qual, para a Busca Tabu, pode ou não ser um mínimo local.

O capítulo seguinte descreve as metodologias estudadas, conceitos fundamentais utilizados pela Busca Tabu e a heurística hibrida proposta e implementada

neste trabalho.

Capítulo 2

Metodologias

Esta seção apresenta o modelo matemático para o problema MSSC, os dois métodos não-hierárquicos do estudo, K-Means e a H-Means e a versão do método Busca Tabu, estudada e implementada neste trabalho. Outros métodos são apresentados neste capítulo. Os métodos são o HK-Means, que seria uma junção de H-Means e K-Means, e o outro método é o K.Base Tabu, a qual é a junção de 2 métodos do estudo: K-Means e Busca Tabu.

2.1 Modelo Matemático

Seja $\mathbf{X} = ((\mathbf{x_1}, ..., \mathbf{x_N})$ um conjunto de dados, em que $\mathbf{x_j} = (\mathbf{x_{1j}}, ..., \mathbf{x_{pj}})$ sendo N observações (ou pontos) no espaço Euclidiano R^p. Considere D subconjuntos disjuntos (grupos) G_i de $\mathbf{X}$. O objetivo dos métodos K-Means e H-Means, já mencionado na seção Introdução, é minimizar o MSSC ou a variabilidade. Em outras palavras, seria minimizar a variabilidade de todos os elementos de cada grupo e maximizar as distâncias dos elementos de grupos distintos. Cada um dos dois métodos inicia-se seu algoritmo com uma partição P_D e $\mathbf{X}$ em D subconjuntos G_i aleatória ou não. Considere $\mathcal{P}_D$ o conjunto de todas as possíveis partições de $\mathbf{X}$ em D subconjuntos. A MSSC (Minimum Sum of Squares Clustering) é a seguinte:

$$MSSC = \min_{P_D \in \mathcal{P}_D} \sum_{i=1}^{D} \sum_{\mathbf{x}_l \in G_i} || \mathbf{x}_l - \bar{\mathbf{x}}_i ||^2, \tag{2.1}$$

em que $||.||$ denota a norma da distância euclidiana.

O problema de minimizar a soma de quadrados interna dos clusters consiste em encontrar uma partição P_D de $\mathbf{X}$ em D subconjuntos disjuntos G_i tais que a soma de quadrados de cada elemento $\mathbf{x}_l$ pertence G_i para a sua centróide $\bar{\mathbf{x}}_i$ é mínima.

2.2 K-Means

MacQueen (1967) sugeriu o termo *K-Means* para descrever um algoritmo que atribui cada elemento ao *cluster* com o centroide mais próximo. Basicamente, o algoritmo é constituído pelas três etapas descritas a seguir.

Passo 1: Sejam G_i $(i = 1, \cdots, D)$, uma partição inicial (aleatória ou não) do conjunto $\mathbf{X}$ e $\overline{\mathbf{x}}_i$ sua centroide correspondente.

Passo 2: Atribuir cada elemento (um a um) para a centroide mais próxima e recalcular a centroide para o grupo que recebe o novo indivíduo e para o conjunto que perder o elemento.

Suponha que o elemento $\mathbf{x}_j$ que pertence ao *cluster* G_l é transferido para outro *cluster* G_i $(l \neq i)$. Johnson e Wichern (2002) apresentam que as centroides desses novos grupos podem ser obtidas a partir das seguintes expressões

$$\overline{\mathbf{x}}_l = \frac{n_l \overline{\mathbf{x}}_l - \mathbf{x}_j}{n_l - 1} \quad e \quad \overline{\mathbf{x}}_i = \frac{n_i \overline{\mathbf{x}}_i + \mathbf{x}_j}{n_i + 1} \tag{2.2}$$

onde $n_i = |G_i|$ e $n_l = |G_l|$. A mudança no valor da função objetivo causada por este movimento é

$$v_{ij} = \frac{n_i}{n_i + 1} ||\overline{\mathbf{x}}_i - \mathbf{x}_j||^2 - \frac{n_l}{n_l - 1} ||\overline{\mathbf{x}}_l - \mathbf{x}_j||^2.$$

Tais mudanças são computadas para todos os possíveis remanejamentos. Se eles são não-negativos a heurística para com uma partição localmente mínima. Caso contrário, a reatribuição reduzindo mais o valor da função objetivo é executada.

Passo 3: Repita o *Passo 2* até não haver mais reatribuições.

2.3 H-Means

Uma partição inicial $(G_1, G_2, \cdots, G_D)$ é escolhida aleatoriamente e as centroides $\overline{\mathbf{x}}_1, \cdots, \overline{\mathbf{x}}_D$ são calculadas. A heurística *H-Means* pode parar em uma solução degenerada, ou seja, com uma partição tendo menos de M *clusters* não-vazios. Neste caso, será necessária a consideração de um *Passo* adicional no algoritmo. A seguir é descrito o algoritmo.

Passo 1 (Inicialização): Sejam G_i $(i = 1, \cdots, D)$, uma partição inicial do conjunto $\mathbf{X}$ e $\overline{\mathbf{x}}_i$ sua centroide correspondente.

Passo 2 (Atribuição): Atribuir (Alocar) cada elemento $\mathbf{x}_j$ $(j = 1, \cdots, N)$, para a centroide mais próxima $\overline{\mathbf{x}}_i$ $(i = 1, \cdots, D)$.

Passo 3 (Teste de otimização local): Se não houver alteração nas alocações, uma partição ótima local e encontrada. Parar.

Passo 4 (Atualização): Atualize a centroide de cada *cluster* e volte ao *Passo*

2.

Caso degenerado

Passo 3' (Teste de otimização local): Se existir mudança na atribuição ir para o *Passo 4*. Caso contrário, a partição local ótima é encontrada. Se for adequado, pare. Se possuir t grupos degenerados vazios, selecione os t elemento mais distantes de suas centroides e inserir na solução como grupos de único elemento e voltar ao *Passo 2*.

Passo 4' (Atualização): Atualize as centroides $\overline{\mathbf{x}}_i$ de cada *cluster* G_i e volte para o *Passo 2*.

Em Hansen e Mladenovic (2001) a heurística HK-Means representa a utilização da solução obtida em H-Means como partição inicial da heurística K-Means (utilizando o mesmo tempo máximo de processamento). Entretanto, utilizou-se o termo HK-Means para representar a heurística que toma a solução de K-Means e utiliza como partição inicial em H-Means (utilizando o mesmo número máximo de iterações).

2.4 Princípios da Busca Tabu

A metodologia Busca Tabu é uma heurística que busca explorar regiões distintas do espaço-solução do problema com o objetivo de obter soluções que não sejam apenas mínimos locais. Para tanto, introduz conceitos e estratégias de busca tais como *Diversificação*, *Intensificação*, *Soluções-Elite* e *Lista-Tabu*, dentre outras (ver Glover e Laguna, 2002). Duas muito importantes estratégias da Busca Tabu são a *Diversificação* e a *Intensificação*. A primeira, como o próprio nome define, procura explorar distintas regiões do conjunto-solução do problema, guardando na memória soluções com características especiais encontradas, que são *Soluções-Elite*; a segunda, também como o próprio nome intui, intensifica a busca por melhores soluções na vizinhança de algumas (ou todas) das *Soluções-Elite*.

Na fase da *Diversificação* a busca por regiões distintas dá-se através de perturbações aleatórias realizadas em componentes de determinada solução que possui atrativos (estes a serem definidos conforme o problema em questão), independentemente de ser a melhor solução encontrada até o momento. Já na *Fase da Intensificação* a exploração da vizinhança de uma *Solução-Elite* dá-se com maior rigor e critérios, podendo-se utilizar metodologias já bastante conhecidas e eficientes para a realização de uma busca local. Como a geração de soluções, na primeira estratégia, é feita de forma exaustiva, uma lista de movimentos já realizados deve ser construída de maneira a se evitar ciclagem (looping) do procedimento computacional. Esta lista de

movimentos proibitivos é denominada por *Lista Tabu*, a qual possui tamanho limitado a ser definido como parâmetro da metodologia.

A subseção seguinte apresenta o procedimento algorítmico da heurística formulada e implementada neste trabalho.

2.5 Heurística BaseTabu

A metodologia proposta neste trabalho, denominada BaseTabu, baseia-se nos princípios da Busca Tabu, acima citados, e em modificações realizadas no algoritmo apresentado por Al Sultan (1995) [2]. Na subseção seguinte, o algoritmo é apresentado.

2.5.1 Algoritmo H.BaseTabu

O algoritmo da heurística BaseTabu é denominado neste estudo por H.BaseTabu. Ele possui a seguinte descrição:

Sejam A_t, A_c e A_b partições denotadas por partição teste, partição atual (ou corrente) e melhor partição e J_t, J_c e J_b seus respectivos valores de MSSC.

- Passo 1: Seja A_c uma partição arbirtrária e J_c o valor de MSSC correspondente descrito em (2.1).

  Faça $A_b = A_c$ e $J_b = J_c$, selecione os valores para os seguintes parâmetros: NTS, número de soluções geradas na *Fase de Diversificação* (ou número de perturbações) e $ITERMAX$, o número máximo de iterações. Seja $k = 1$, e vá para o Passo 2.

- Passo 2: **(Fase da Diversificação)**

  Gere NTS soluções testes A_t^1, A_t^2, ..., A_t^{NTS} e avalie as funções objetivos correspondentes J_t^1, J_t^2,..., J_t^{NTS}, onde A_t^i é obtida de A_c trocando-se aleatoriamente a alocação de i elementos para i grupos, $i = 1, 2, ..., NTS$. Vá para o Passo 3.

- Passo 3: De forma crescente, ordene J_t^1, J_t^2, J_t^3,..., J_t^{NTS} e denote-as por $J_t^{[1]}$, $J_t^{[2]}$,..., $J_t^{[NTS]}$.

  Se $J_t^{[1]} < J_b$, então faça $A_b = A_t^{[1]}$ e $J_b = J_t^{[1]}$, e uma busca intensiva aplicando o algoritmo H.Means tendo como solução inicial a partição A_b. Faça $k = k + 1$ e volte para o Passo 2.

  Caso contrário seja $A_c = A_t^{[1]}$, $J_c = J_t^{[1]}$ Solução-Elite. Faça $k = k + 1$. Se $k \geq ITERMAX$, vá para o Passo 4; caso contrário, volte para o Passo 2.

- Passo 4: **(Fase da Intensificação)**

Sejam $A_s^{[1]}$, $A_s^{[2]}$, $A_s^{[3]}$,...,$A_s^{[INT]}$ Soluções-Elite com valores de MSSC iguais a $J_s^{[1]}$, $J_s^{[2]}$,..., $J_s^{[INT]}$, respectivamente.

Seja J_s^{MIN} o valor mínimo de $J_s^{[1]}$, $J_s^{[2]}$,..., $J_s^{[INT]}$, e A_s^{MIN} a partição correspondente. Aplicar o algoritmo H.Means com A_s^{MIN} como partição inicial.

- Passo 5: Seja A_f e J_f a solução obtida no passo anterior. Se $J_f < J_b$, então $J_f = J_b$ e $A_f = A_b$. A_b é a melhor solução encontrada e J_b é o melhor valor da função objetivo correspondente. Caso contrário, J_b é a melhor solução.

  Repetir os Passos 4 e 5 para no máximo INT vezes.

Note que, no Passo 3, uma *Intensificação* é realizada se uma solução encontrada é melhor que o mínimo até o momento. Se esta *Intensificação* não acontece, então a melhor solução gerada na *Fase de Diversificação* (Passo 2) é selecionada e guardada como *Solução-Elite*.

A fase de maior *Intensificação* do algoritmo compreende os Passos 4 e 5, e nela as regiões eleitas são aquelas que são vizinhanças das *Soluções-Elite* com menor MSSC.

2.6 HK-Means

Esse método é a união dos dois métodos Não-Hierárquicos do estudo: K-Means e H-Means. A partição inicial do algoritmo é dada pelo método K-Means. O resto do processo é feito pelo método H-Means.

2.7 K.BaseTabu

Nesta metodologia, os três métodos do trabalho são incorporados em um mesmo algoritmo. Na H.BaseTabu, o primeiro passo é uma partição aleatória. No K.BaseTabu, essa partição inicial é dada pelo do método K-Means, com o máximo de iterações do algoritmo. Depois desse passo, o resto do algoritmo é o mesmo para o algoritmo H.BaseTabu.

Capítulo 3

Resultados e Discussão

Esta seção apresenta os resultados computacionais obtidos e a análise comparativa entre os métodos descritos no capítulo anterior. Para tanto, utilizou-se o algoritmo K-means disponível no sistema R. O algoritmo do H-Means e HK-Means forma implementados nesse mesmo sistema, e disponibilizados por Quintino F. (2015). Os algoritmos H.BaseTabu e K.BaseTabu foram implementados pelo próprio autor do trabalho. Suas implementações foram realizadas em linguagem compatível à do Sistema de Computação Estatística R. A implementação do algoritmo H.BaseTabu é apresentada no Apêndice A.

Os resultados desse estudo foram obtidos através de testes com banco de dados do próprio sistema de computação estatística R. Esses bancos de dados são: USArrests e Iris. Todos os métodos mencionados na seção anterior foram testados e os resultados obtidos são apresentados com detalhes nas próximas subseções.

3.1 Exemplo USArrests

Este banco de dados consiste em cinquenta observações, as quais são referentes aos estados dos Estados Unidos. Cada observação possui quatro variáveis. As variáveis são porcentagem de assassinato, assalto, estupro e população urbana. Todas as variáveis são quantitativas.

Com esse banco de dados, testou-se todos os 5 métodos do estudo. Cada algoritmo, realizou 15 iterações. Nos métodos que necessitam de perturbações na *Fase de Diversificação*, como o H.BaseTabu e o K.BaseTabu, foram realizadas 10 e 20 perturbações, como mostram as Tabelas 3.1, 3.2 e 3.3 abaixo.

Tabela 3.1: Valores de MSSC obtidos pelos métodos para diferentes número de grupos (D), com 15 iterações e 10 perturbações, USArrests

D	Métodos				
	K-Means	H-Means	HK-Means	H.BaseTabu	K.BaseTabu
2	96399.03	96399.03	96399.03	96399.03	96399.03
3	47964.27	47964.27	47964.27	47964.27	47964.27
4	37652.66	41475.87	38731.8	41475.87	37730.8
5	24417.02	29079.61	24495.17	28240.23	32524.3
6	22290.84	29272.64	26035.9	19111.84	22649.99
7	16563.67	25468.34	25269.89	18271.78	19723.35
8	13259.15	24111.56	13259.15	17264.79	14459.58

Fonte: Sistema de computação estatística R - USArrests

A tabela acima apresenta os valores de MSSC obtidos pelos métodos para diferentes número de grupos, com 15 iterações e 10 perturbações. Por essa tabela, observa-se que todos os métodos obtém mesmas soluções, para o número de grupos 2 e 3. A partir de 4 grupos, essa igualdade não acontece mais entre eles. Percebe-se que a medida que se aumenta o número de grupos, os valores das funções objetivos de cada metodologia diminui. Há casos isolados, em que as funções objetivos aumentam, mas em pouquíssimos casos. Alguns deles, como no número de grupos de 5 para 6, que o valor de MSSC do método HK-Means, passa de 24495,17 para 26035,9 ou no método H.Means, que no número de grupos de 5 para 6, o seu valor de MSSC era de 29079,61 e sobe para 29272,64 (isto pode ser explicado por causa da aleatoriedade da partição inicial dos algoritmos).

Por essa mesma tabela, é importante verificar essa redução dos valores de MSSC em todos os métodos a medida que se aumenta o número de grupos. E isso leva a um ponto favorável da implementação, no sistema de computação estatística R, dos algoritmos H.Means, HK-Means, H.BaseTabu e K.BaseTabu, pois, é um comportamento esperado.

Outro fator a considerar é a adequabilidade do método H.BaseTabu. Este algoritmo, como já foi mencionado, é uma junção do método H.Means e o método BaseTabu. Ao comparar H.Means com o método H.BaseTabu, pode se observar que o H.BaseTabu tem um desempenho bem melhor. Para números de grupos iguais a 2 e 3, os valores das MSSC dos dois métodos são os mesmos. Porém, a partir do número de grupo igual a 4, o método H.BaseTabu sempre é melhor, comparado ao método H.Means. Existem casos com grande diferença de valores de MSSC, como no número de grupos igual a 6, o qual o valor de MSSC de H.Means é 29272.64, enquanto, em H.BaseTabu, é 19111.84. Outro caso é para o número de grupos igual a 8, no qual o

método H.Means possui o valor de MSSC igual a 24111.56 e o H.BaseTabu, 17264.79. Logo, pela Tabela 3.1, o método H.BaseTabu tem um desempenho bem melhor que o do método H.Means.

De todos os métodos em estudo, um outro ponto a ser destacado é que o método H.BaseTabu tem valores mais próximos ou até melhores que o método K.Means. E esse fator é favorável à metodologia H.BaseTabu, pois, observando de modo geral por Tabela 3.1, e fazendo comparações de todos esses métodos, dois a dois, para todos os números de grupos, o melhor método é o K.Means. Nesta tabela, H.BaseTabu é o único método que tem o valor de MSSC menor que o método K.means, como se pode observar para o número de grupos igual a 6. Porém, procurando conhecer o segundo método em ordem de desempenho, nota-se um equilíbrio entre os métodos HK-Means, H.BaseTabu e K.BaseTabu.

Estes três últimos métodos, quando comparados ao método K-Means, percebe-se que o método mais próximo é o H.BaseTabu, com destaque ao número de grupos igual a 6, o qual esse método é melhor que o K.Means. Os outros métodos só têm valores iguais ou maiores que K-Means. O método HK-Means é mais próximo de K-Means que os outros dois métodos, K.BaseTabu e H.Means, e isso pode ser observado nos casos em que o número de grupos é igual a D=4, D=5, D=6, D=7 e D=8.

Em seguida, o quarto melhor método é o método K.BaseTabu. O método com pior desempenho é o método H.Means.

Tabela 3.2: Valores de MSSC obtidos pelos métodos para diferentes número de grupos (D), com 15 iterações e 20 perturbações, USArrests

D	Métodos				
	K-Means	H-Means	HK-Means	H.BaseTabu	K.BaseTabu
2	96399.03	96399.03	96399.03	96399.03	96399.03
3	47964.27	47964.27	47964.27	47964.27	47964.27
4	37652.66	41475.87	38731.8	36989.6	37652.66
5	24417.02	29079.61	24495.17	28240.23	24417.02
6	22290.84	29272.64	26035.9	19111.84	18768
7	16563.67	25468.34	25269.89	18107.95	16563.67
8	13259.15	24111.56	13259.15	17044.17	13927.75

Fonte: Sistema de computação estatística R - USArrests

A Tabela 3.2 apresenta os valores de MSSC obtidos pelos métodos em estudo, com uma diferença: Os métodos H.BaseTabu e K.BaseTabu foram testados com 20 perturbações na *Fase da Diversificação*.

Nesta tabela, ainda tem-se a mesma situação da Tabela 3.1, para o número de grupos igual a 2 e 3. Os valores de MSSC tem os valores iguais nesses números de

grupos. Porém, uma diferença em relação a tabela anterior, é que comparando-se os métodos K.Means e K.BaseTabu, percebe-se um fato interessante. O método K-Means, ainda tem um bom desempenho. Porém, o melhor método entre os dois, é o método K-BaseTabu. Esses métodos possuem valores de MSSC iguais até o número de grupos igual a 5. Mas para o número de grupos igual a 6, o valor de MSSC de K.BaseTabu é menor que o valor de K.Means em 16 %. O único caso em que o método K.Means é melhor que o K.BaseTabu, pela Tabela 3.2, é no número de grupos igual a D=8. Entretanto a diferença entre os valores de MSSC dos dois métodos é muito baixa.

Comparando-se os outros métodos com o K.BaseTabu, pela Tabela 3.2, nota-se que o método K.BaseTabu tem o melhor desempenho. Em poucas situações, o método K.BaseTabu é pior que esses outros métodos. Um desses casos é para o número de grupos igual a 4, que comparando-o ao método H.BaseTabu, o valor de MSSC de K.BaseTabu é maior 667.06, em números absolutos. Outro caso é para o número de grupos igual a 8, o qual os métodos HK.means e K.Means têm seus valores de MSSC menores que o valor do método K.BaseTabu.

Estes resultados favoráveis ao método K.BaseTabu pode ser explicado pelo fato do mesmo inicializar o procedimento algorítmico utilizando a solução do K-Means e pelo aumento no número de perturbações na *Fase da Diversificação*.

Em seguida, o método H.BaseTabu é o terceiro em ordem de desempenho. O método HK-Means é o quarto, e o último método novamente, é o método H.Means.

Para se observar melhor o comportamento do algoritmo H.BaseTabu, a Tabela 3.3 abaixo, apresenta resultados obtidos antes (J_b antes) e depois (J_b final) da Fase de Intensificação. Como esse algoritmo foi testado com 10 e 20 perturbações, comparou-se os valores de MSSC em cada uma dessas situações.

Tabela 3.3: Comparação do valor de MSSC antes e ao final da *Fase de Intensificação* do Método H.BaseTabu para 10 e 20 perturbações, 15 iterações, USArrests

D	10 Perturbações		20 Perturbações	
	J_b Antes	J_b Final	J_b Antes	J_b Final
2	96399.03	96399.03	96399.03	96399.03
3	47964.27	47964.27	47964.27	47964.27
4	41475.87	41475.87	41332.15	36989.6
5	40489.45	28240.23	29079.61	28240.23
6	19120.24	19111.84	28075.71	19111.84
7	18952.04	18271.78	25468.34	18107.95
8	17804.33	17264.79	18037.65	17044.17

Fonte: Sistema de computação estatística R - USArrests

Uma observação importante é que o "antes"significa o melhor valor da

MSSC após todas as iterações e perturbações, só que sem ter passado pelo processo da intensificação, que é o Passo 4 do algoritmo. Essa parte da intensificação foi implementada no algoritmo H.BaseTabu com o intuito de procurar diminuir mais o valor da função objetivo. Logo, o J_b final representa o valor da MSSC após o final dessa intensificação.

Observando a Tabela 3.3, percebe-se que os valores, tanto inicial como final, são iguais para o número de grupos D=2 e D=3, independente do número de perturbações. Porém, essa igualdade não permanece a mesma, mostrando ser as duas estratégias, *Diversificação* e *Intersificação*, essenciais para a eficiência do método. Ou seja, pode-se notar que para 20 perturbações a *Fase da Intensificação* foi mais efetiva. Contudo, em alguns casos, os valores de MSSC finais são iguais para o número de grupos iguais a D=5 e D=6, e são quase os mesmos valores para os demais número de grupos.

Olhando para a transição dos valores J_b *Antes* e J_b *Final*, tanto com 10, como com 20 perturbações, percebe-se que há uma diminuição desses valores. Se observar essa transição, separando os valores para 10 e 20 perturbações, constata-se um ponto relevante. Para 10 perturbações, o valor reduziu muito num caso isolado, o caso com número de grupos igual a 5. Em contrapartida, nos outros casos, diminuiu-se muito pouco ou não diminuiu. Com 20 perturbações, percebe-se que a partir do número de grupos iguais a D=4, os valores de MSSC são melhores ao final, o que sugere a importância da *Fase da Diversificação* para a metodologia.

3.2 Exemplo Iris

O banco de dados Iris é composto por 150 observações, as quais são referentes a espécies de plantas. Essas plantas são classificadas por três espécies. As espécies são setosa, versicolor e virginica. Cada observação ou elemento, tem cinco variáveis. Dentre as variáveis, 4 são classificadas como variáveis quantitativas e uma qualitativa. A variável qualitativa é a variável que classifica as espécies. Tal variável não entra na análise de clusters em estudo. As quatro variáveis quantitativas são: Largura da pétala, Largura da sépala, Comprimento da pétala e Comprimento da sépala.

Com esse banco de dados, testes computacionais foram realizados entre os métodos em estudo neste trabalho. Nestes, os métodos H.BaseTabu e K.BaseTabu foram testados, na *Fase da Diversificação*, com 20 e 30 perturbações. Todos os algoritmos foram testados com número de iterações igual a 15. As Tabelas 3.4, 3.5 e 3.6 mostram os resultados obtidos.

Tabela 3.4: Diferenças entre a Soma de Quadrados Miníma dos Clusters e a solução ótima, segundo o número de clusters, para o banco de dados Íris, 15 iterações e 20 perturbações

D	Valor Ótimo	Desvio da solução ótima em 15 iterações e 20 perturbações				
		K-Means	H-Means	HK-Means	H.BaseTabu	K.BaseTabu
2	152.3470	0.00	0.00	0.00	0.00	0.00
3	78.8525	0.00	0.00	0.00	0.00	0.00
4	57.2284	0.04	0.00	0.00	0.00	14.22
5	46.4461	3.38	10.81	3.38	6.46	3.6
6	39.0399	9.04	3.38	6.03	6.56	6.48
7	34.2982	9.28	9.4	3.3	5.8	4.21
8	29.9889	2.88	13.25	8.53	3.96	2.57
9	27.7860	6.78	14.66	3.79	6.44	0.2
10	25.8340	0.68	8.74	5.09	5.45	0.00
Erro médio		3.56	6.69	3.35	3.85	3.48

Fonte: Sistema de computação estatística R - iris

Nessa tabela, comparou-se todos os métodos, com 15 iterações, e 20 perturbações, no caso do H.BaseTabu e K-BaseTabu. Os valores ótimos da função objetivo foram extraídos do artigo dos autores Hansen e Mladenovic (2001). Logo, com esses valores ótimos, foi feito a comparação pelos desvios e os erros médios de cada método. Pelos desvios, percebe-se que os valores da função objetivo, ou MSSC, são os mesmos para todos os métodos no número de grupos iguais a D=2 e D=3. Porém, percebe-se que esse comportamento de igualdade não é mais o mesmo no resto do processo.

Através dos resultados representados na Tabela 3.4, pode-se perceber que o método com o pior desempenho é o método H.Means, (exceto para D=6, o qual seu desvio é de 3.38). Seus desvios são os piores em relação aos outros métodos, porque apresenta desvios que passam de 10.00, como se percebe nos números de grupos igual a D=5, D=8 e D=9. Com isso, o erro médio do método H.Means é o maior em relação aos outros métodos.

Verificando os desvios dos outros 4 métodos, os métodos K.Means e HK-Means seguem um mesmo comportamento. Por exemplo, o K-Means tem um desvio baixo para o número de grupos igual a D=5, enquanto para o número de grupos igual a D=6, o valor do desvio aumenta muito. Isso se repete para o número de grupos igual a D=8, o qual o valor do desvio é 2.88 e o desvio do número de grupos igual a D=9 é 6.78, observando novamente um aumento considerável. O método HK.Means possui um desvio baixo para o número de grupos igual a D=5, o qual o valor do desvio é 3.38 e o desvio do número de grupos igual a D=6 é 6.03. Novamente isso acontece para o

número de grupos igual a D=7, o qual o valor do desvio é 3.3, enquanto para o número de grupos igual a D=8 é 8.53, constatando mais uma vez um crescimento considerável.

Um ponto importante é que pode-se observar que o método H.BaseTabu possui desvios bem menores do que o H.Means, como nota-se nos números de grupos iguais a D=5, D=8, D=9 e D=10. Um fator relevante também é que o método H.BaseTabu tem desvios com valores menores do que o método K.Means e em vários casos como nos números de grupos iguais a D=4, D=6, D=7 e D=9.

Um método que também teve um bom desempenho foi o método K.BaseTabu, o qual a partição inicial é a partição do método K.Means. O método K.BaseTabu teve apenas um caso isolado de todos os seus valores de desvios, que é o desvio obtido para o número de grupos igual a D=4. Esse valor é 14.22. Como se verifica no restante dos desvios, os valores diminuem muito, e chegam a desvios próximos de 0 ou igual a 0, como número de grupos igual a D=9 e D=10, apresentando que o algoritmo tem boas soluções. Além disso, vários valores de desvios do mesmo foram menores do que os desvios do método K.Means. Esses casos são apresentados nos números de grupos iguais a D=6, D=7, D=8, D=9 e D=10. Assim, tanto pelos desvios, como pelo erro médio, percebe-se que o método K.BaseTabu tem um desempenho melhor que o método K.Means.

Pelos erros médios, o menor erro médio foi o do método HK-Means. O valor do erro médio é 3.35. Por essa medida, o método HK-Means deveria ser considerado o método com o melhor desempenho dos 5 métodos exibidos na Tabela 3.4. Contudo, observando os valores dos desvios para todos os números de grupos, e pelo erro médio, o método K.BaseTabu é o método que tem o melhor desempenho de todos. Pois embora tenha tido um caso com um desvio muito alto (como é apresentado no número de grupos igual a D=4), percebe-se que alguns valores de desvios do método K.BaseTabu são menores que os desvios de HK-Means. Esses valores são mostrados para os números de grupos iguais a D=8, D=9 e D=10. Mais um ponto importante nessa comparação é que os desvios finais do método K.BaseTabu, como para os números de grupos iguais a D=8, D=9 e D=10, tendem a zero, diferentemente dos desvios finais de HK.Means. O valor do desvio do método K.BaseTabu para o número de grupo igual a D=10 é 0.00. Esse ponto é importante, pois para o maior número de grupos, o algoritmo consegue o valor ótimo do problema MSSC. Além disso, só esse método tem um desvio nulo para o número de grupos igual a D=10.

Assim, em ordem de desempenho, o primeiro mais eficiente é o método K.BaseTabu, o segundo, o método HK.Means. O terceiro, o método K.Means. O quarto método é o H.BaseTabu e o pior método é o H.Means.

Tabela 3.5: Diferenças entre a Soma de Quadrados Miníma dos Clusters e a solução ótima, segundo o número de clusters (D), para o banco de dados Íris, 15 iterações e 30 perturbações

D	Valor Ótimo	Desvio da solução ótima em 15 iterações e 30 perturbações				
		K-Means	H-Means	HK-Means	H.BaseTabu	K.BaseTabu
2	152.3470	0.00	0.00	0.00	0.00	0.00
3	78.8525	0.00	0.00	0.00	0.00	0.00
4	57.2284	0.04	0.00	0.00	0.00	0.04
5	46.4461	3.38	10.81	3.38	6.46	3.41
6	39.0399	9.04	3.38	6.03	3.37	8.62
7	34.2982	9.28	9.4	3.3	9.4	3.10
8	29.9889	2.88	13.25	8.53	5.77	9.25
9	27.7860	6.78	14.66	3.79	7.97	0.25
10	25.8340	0.68	8.74	5.09	7.97	3.50
Erro médio		3.56	6.69	3.35	4.55	3.13

Fonte: Sistema de computação estatística R - iris

A Tabela 3.5 apresenta resultados obtidos, com o mesmo banco de dados, para testes, com 30 perturbações realizadas na *Fase da Diversificação* dos métodos H.BaseTabu e K.BaseTabu.

Para o número de grupos igual a D=2 e D=3, os valores de MSSC de todos os métodos são os mesmos, em relação ao valor ótimo. Os desvios começam realmente no número de grupos igual a D=4, porém, apenas no K-Means e no K.BaseTabu, que seus valores se diferem minimamente do valor ótimo. A partir do número de grupos igual a D=5, os desvios dos algoritmos tendem a ser mais diferentes entre si, e diferentes também do valor ótimo.

Nota-se que o método H-Means continua tendo o pior desempenho, com um erro médio bem acima dos demais. Em sequência, o método H.BaseTabu aumenta seu erro médio, piorando assim seu desempenho. Porém, seus desvios ainda continuam a ser bem menores que os desvios do método H.Means, como se apresenta nos números de grupos iguais a D=5, D=6, D=8, D=9 e D=10. Outro ponto importante do método H.BaseTabu é que em alguns casos, os desvios do mesmo foram menores do que K-Means. Isso é apresentando no número de grupos de D=4 e D=6 da Tabela 3.5.

Comparando os métodos HK.Means e o K.BaseTabu, o método K.BaseTabu continua a ter alguns desvios menores em relação ao método HK.Means. Esses casos são mostrados nos números de grupos iguais a D=7, D=9 e D=10. Além disso, o erro médio de K.BaseTabu é menor do que o valor do erro médio de HK.Means. Essas duas situações se repetem quando se compara o método K.BaseTabu com o método

K.Means. Os valores de desvios do método K.BaseTabu são menores do que os valores de K.Means em alguns momentos. Como para os números de grupos iguais a D=6, D=7 e D=9. E o erro médio do K.BaseTabu é menor do que do método K.Means.

Logo, por todas essas comparações, percebe-se que o método com melhor desempenho, pela Tabela 3.5, é o método K.BaseTabu. O segundo melhor método é o HK.Means. E o terceiro o método K.Means. Como já foi mencionado, o quarto método é o H.BaseTabu e o método com o pior desempenho é o método H.Means.

A Tabela 3.6, abaixo, apresenta os valores de MSSC obtidos antes e após a a realização da *Fase de Intensificação* dos algoritmos H.BaseTabu e K.BaseTabu nos testes realizados e apresentados nas Tabelas 3.4 e 3.5.

Tabela 3.6: Comparação do valor de MSSC antes e ao final da Fase de Intensificação do Método H.BaseTabu para 20 e 30 perturbações, 15 iterações, iris

D	20 Perturbações		30 Perturbações	
	J_b Antes	J_b Final	J_b Antes	J_b Final
2	152.348	152.348	152.348	152.348
3	78.85567	78.85567	78.85144	78.85144
4	57.25601	57.22847	57.25601	57.22847
5	52.99335	52.90907	52.94464	52.90178
6	45.59262	45.59262	42.40302	42.40302
7	40.09963	40.09963	45.425	43.69679
8	34.08542	33.94242	35.75025	35.75025
9	34.31277	34.22944	35.75025	35.75025
10	31.28823	31.28823	34.44495	33.80783

Fonte: Sistema de computação estatística R - iris

Constata-se que todos os valores de MSSC encontrados para o número de grupo igual a D=2 são iguais. No número de grupos igual a D=3, os valores inicial e final, são iguais, dentro das suas respectivas perturbações.

A partir do número de grupos igual a D=4, o comportamento varia em alguns aspectos. Para 20 perturbações, a *Fase da Intensificação* é pouco efetiva dependendo do número de grupos. Para os números de grupos iguais a D=5, D=8 e D=9, nota-se melhores resultados para J_b *Final*, porém, essa melhora é pequena.

Para 30 perturbações, a *Fase da Intensificação* melhora os resultados mais do que na fase com 20 perturbações. Observa-se essa melhora nos casos para os números de grupos iguais a D=4, D=5, D=7 e D=10. Outro ponto para essa melhora para 30 perturbações, é que no número de grupos igual a D=7, a *Fase da Intensificação* faz com o valor de MSSC diminua significativamente, comparado a todas as outras fases de cada número de grupos, tanto para 20 perturbações, como para 30 perturbações.

Assim, pelas Tabelas 3.3 e 3.6, percebe-se que a medida que se aumenta o número de perturbações na *Fase da Diversificação*, os resultados finais melhoram. Em algumas situações, com diferenças mais significativas, em outros casos, nem tanto.

Nos testes realizados optou-se por apenas duas *Intensificações* (Passos 4 e 5 do algoritmo H.BaseTabu). Os resultados obtidos sugerem para a necessidade de se aumentar este número, explorando regiões de um número maior de *Soluções-Elite*.

Capítulo 4

Conclusões e Considerações Finais

Esse trabalho teve como objetivo principal formular e implementar uma heurística híbrida, que é a união da metodologia Busca Tabu (ou Tabu Search) e a heurística H-Means, todas voltadas para a resolução do problema MSSC (Minimum Sum of Squares for Clustering), em linguagem compatível à do sistema de computação estatística R. Tal objetivo foi conquistado.

Outros objetivos específicos como o aprendizado do aluno na utilização do sistema de computação R, foi alcançado. Outro objetivo específico era de se obter boa adaptação da heurística proposta ao ambiente estatístico R e que as soluções indicadas por esta versão fossem tão boas ou melhores quanto aquelas indicadas pelos métodos não-hierárquicos K-Means, disponível no mesmo sistema computacional, e H-Means, este último recentemente programado em linguagem R [Felipe Quintino, PIBIC 2014/2015]. Com os dois exemplos, USArrests e Iris, bancos de dados do sistema de computação estatística do R, descritos no capítulo anterior, notou-se que em parte esse último objetivo específico foi cumprido.

Pelas Tabelas 3.1, 3.2, 3.4 e 3.5, percebeu-se que o método com pior desempenho foi o método H.Means. O H.BaseTabu em quase todos os casos dos números de grupos, apresentou melhor desempenho do que o H.Means (mais detalhes, volte ao capítulo anterior). Logo, o objetivo de se obter uma boa adaptação da heurística implementada no sistema R e que tivesse soluções tão boas ou melhores do que as soluções indicadas por H.Means foi alcançado. Porém, isso não aconteceu do mesmo jeito com relação ao método K-Means. Contudo, verificando-se as Tabelas 3.2, 3.4 e 3.5, nota-se que várias soluções de H.BaseTabu são tão boas ou até melhores do que as soluções de K-Means, ao qual esse fato é perceptível principalmente na Tabela 3.4.

Dois pontos a serem ressaltados da heurística BaseTabu são a *Fase da Intensificação* e sua adaptação ao algoritmo H.BaseTabu. A *Fase da Intensificação* atua em partições encontradas da *Fase da Diversificação* (como descrito na subseção 2.5.1). Pelas Tabelas 3.3 e 3.6, constatou-se que aumentando o número de perturbações da

Fase da Diversificação, os resultados dos valores de MSSC são melhorados.

Um resultado importante para o trabalho aqui apresentado é que o algoritmo H.BaseTabu, quando aplicado a solução obtida pelo K.Means, conseguiu obter melhora do valor de MSSC em número de grupos igual a D=6 da Tabela 3.2, números de grupos iguais a D=6, D=7, D=8, D=9 e D=10 da Tabela 3.4 e números de grupos iguais a D=6, D=7 e D=9 da Tabela 3.5.

Este fato comprova a adequação da implementação da metodologia proposta.

Uma consideração importante a ser feita é que os algoritmos H.Means e H.BaseTabu são bons algoritmos, mas podem ser melhorados. Algumas adaptações, devem ser feitas nos mesmos para esta melhoria. No algoritmo H.BaseTabu, mais *Fases de Intensificações* seria uma maneira de melhorar a solução ótima. Outra consideração nesse algoritmo, é que se ao invés de intensificar a partição utilizando o método H.Means, para tanto aplicar método K-Means.

Para aumentar a eficiência e a adequação do método H.BaseTabu proposto, melhorar a programação do algoritmo no sistema de computação estatística R e fazer testes com bancos de dados maiores são passos a serem realizados.

Referências Bibliográficas

[1] Aloise, D., Hansen, P., *Clustering: A Chapter for Handbook for Discrete and Combinatorial Optimization*, Les Cahiers du GERAD, April, (2008).

[2] Al-Sultan, K., *A Tabu search approach to the clustering problem*, Pattern Recognit, vol. 28, no. 9, pp 1443-1451, (1995).

[3] Babu, G. e Murty, M., *A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm*, Pattern Recognit, Lett., vol. 14, no. 10, pp 763-769, (1993).

[4] Biagio, M. A., *A Recovering Comparative Study of Clustering Analysis*, Tendências em Matemática Aplicada e Computacional, 1, n.2, pp 303-317, (2000).

[5] Brusco, M.J., Steinley, D., *A Comparison of Heuristic Procedures for Minimum Within-Cluster Sums of Squares Partitioning*, Psychometrika, vol.72, 4, pp 583-600, (2007).

[6] Everitt, B., *Cluster Analysis*, London, Heinemann Ed. Books, (1993).

[7] França, P.M., Sousa, N.M., Pureza, V., An Adaptive Tabu Search Algorithm for the Capacitated Clustering Problem, Internacional Transactions in Operational Research, Volume 6, pp 655-678, (1999).

[8] Glover, F., Laguna, M., *Tabu search*, Kluwer Academic Publishers, (2002).

[9] Hansen, P., Mladenovic N., *J-MEANS: a new local search heuristic for minimum sum of squares clustering*, Pattern Recognition 34, pp 405-413, (2001).

[10] Hansen, P., Mladenovic N., *Variable Neighborhood Search*, In: Burke, E., Kendall, G.; Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques; Springer Science, N.Y., pp 211-238, (2005).

[11] Jain, A. K., *Data Clustering: 50 Years Beyond K-means*, Pattern Recogn, Lett., 31, pp 651-666, (2010).

[12] Johnson, R. A. e Wichern, D. W., *Applied Multivariate Statistical Analysis*, sexta edição, pp 671-715, (2007).

[13] Kaufman, L. e Rousseuw, P., *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York, (2005).

[14] Ng, M. K. e Wong, J. C., *Clustering Categorical Data Sets Using Tabu Search Techniques*, Pattern Recognition vol. 35, pp 2783-2790, (2002).

[15] Parsha, M. K. e Pacha, S., *Recent Advances in Clustering Algorithms: A Review, Int. J. of Conceptions on Computing and Information Technology*, vol.1, Issue 1., Nov. 2013, (2013).

[16] Quintino, F. S., *Metodologia para Agrupamento de Dados: versão VNS para Sistema R*, PIBIC 2014-2015, trabalho a ser apresentado no Congresso de Iniciação Científica, (Brasília, DF), 2015.

Apêndice A

Códigos da versão H.BaseTabu

Neste apêndice apresenta os códigos desenvolvidos em linguagem R de computação estatística da versão do Algoritmo H.BaseTabu do Capítulo 2.

```
H.BaseTabu<- function(elementos,M,k,np,r,iter.max=15){

  elementos
  dados<-as.matrix(elementos)
  r<- ncol(dados)+1          # r é o número de colunas mais um #
  n<- nrow(dados)            # n é o número de linha das matriz #
  k                          # número de perturbações #
  np                         # número funções objetivos das perturbações #
  iter.max                   # número máximo de iterações #
  elit<-0

  # matriz melit #

  ##### ncol2<-ncol(dados) + iter.max
  vmenores<- vector()

  melit <- matrix(nrow=n, ncol=iter.max)
  ####   v.menores <-  c(1:iter.max)

  # Condição para que a matriz seja validada #

  if(is.matrix(dados)){
    dados<-as.data.frame(dados)
  }

  ########  Passo 1  ##########

  # Partição Inicial #
```

```
dados[,r] <- sample(1:M,n,replace=TRUE)


### Função Perturbação ###


Perturbacao <- function(dados, k, r, M){
  dados2 <- dados
  # Escolhendo a posicao aleatoriamente #
  # Escolhendo aleatoriamente os clusters para as posicoes #
  vet<-as.vector(dados[,r])
  mat<-matrix(,ncol=length(dados[,r]),nrow=k)
  vet0<-vet
  for(j in 1:k){
    a<-sample(1:length(dados[,1]), j)
    vet<-vet0
    vet[a] <- sample(1:M, j, replace=TRUE)

    mat[j,]<-as.factor(vet)
  }

  final<- list(dados=dados2,
               pert = t(mat),
               com = cbind(dados2,t(mat)))
  return(final)
}


#### Função Objetivo para a partição inicial  #####

##  Função objetivo atual   ##

f.atual2 <- function(dados, M, r, n){
  #Centroide
  centro <<- matrix(0,nrow = M, ncol = (r-1))
  for (i in 1:(r-1)){
    centro[,i] <- tapply(dados[,i], dados[,r], mean)
  }
  # Distancia entre cada elemento e sua centroide
  distancias <- numeric()
  for (elem1 in 1:n){
    distancias[elem1] <- (dist(rbind(dados[elem1,1:(r-1)],
                                     centro[(dados[elem1,r]), ]),
                               method = "euclidean"))^2
  }
```

```
  f.obj <<- sum(distancias)
  structure(list( centroide = centro,  f.opt = f.obj ))
}

# função objetivo da partição inicial #

jb <- f.atual2(dados,M,r,n)$f.opt
print(jb)


##  Funcao objetivo atual para as perturbações  ##
##  Calcula a funcao objetivo e a centroide ##

objetivo1<-c(1:np)
f.atual <- function(dados, M, r, n, np){

  #Centroide
  for(k in (r+1):(r+np)){
    centro <- matrix(,nrow = M, ncol = (r-1))
    for (i in 1:(r-1)){
      centro[,i] <- tapply(dados[,i], dados[,k], mean)
    }
    # Distancia entre cada elemento e sua centroide
    distancias <- numeric()
    for (elem1 in 1:n){
      m<-rbind(dados[elem1,1:(r-1)],centro[(dados[elem1,k]), ])
      distancias[elem1] <- (dist(m,method = "euclidean"))^2
    }
    objetivo1[k-r]<<-sum(distancias)
  }
}


####### Passo 2 ############
####### Fase da Diversificação ######

iter<-1
for(iter in 1:iter.max) {
perturbacao1<- Perturbacao(dados, k, r, M)
# Perturbacao de M grupos #
perturbacao1
# Matriz dos elementos, partição inicial e M grupos #
dados1<-perturbacao1$com
# funções objetivos de k perturbações #
funcaopertur<- f.atual(dados1, M, r, n, np)
# chamando o vetor dos valores das funções objetivos #
```

```
 objetivo1
 # o menor valor da função objetivo 1 ou jts #
 jt<-min(objetivo1)
 # a menor posição do valor das funções objetivos #
 rt<-which.min(objetivo1)
 # rt é somado com r #
 novapert<-dados1[,(r+rt)]
 # matriz dados com a partição inicial #
 novapert
 # substituindo a partição inicial por rt #
 dados[,r]<-novapert
 dados[,r]
 dados

 ######## Passo 3 #########
 ## Primeira condição desse passo ##

    if( jt < jb) { jb<-jt

    #### Chamar H-Means ####
    # Calcula-se H-Means para a matriz dados #

    resultado<-H.means(dados[,-r], M, iter.max,initial.part = novapert)
    resultado2<- resultado$func.objetivo
    resultado2                  # resultado da função objetivo #
    particaoinicial<-resultado$grupos
    agrupamento<-table(particaoinicial)
    dados[,r]<-particaoinicial
    jb<-resultado2
    }
    else {

####### Passo 4 #######
## FASE DA INTENSIFICAÇÃO ##

    jc<-jt              # matriz de perturbações #
    novapert<-dados1[,(r+rt)]           # rt #
    dados[,r]<-novapert
    dados[,r]

    ## soluções elite ##

    elit<-elit+1
    ##v.menores <- as.vector(elit) ##
    vmenores[elit]<-jc
    melit[,elit] <- novapert
```

```
    }
    iter<-iter + 1
    list(print(jb), print(dados1),print(objetivo1),print(jt),
    print(novapert),print(resultado2),
    print(particaoinicial),print(agrupamento),
    print(vmenores),print(melit))
  }

  ### INTENSIFICAÇÂO EM SOLUÇÕES-ELITE  #######

  int=0

  for (int in 0:1) {

  if( int==0 ) {

  rc <- which.min(vmenores)
  print(rc)
  fc <- vmenores[rc]
  print(fc)
  ultimo <- melit[,rc]

  resultado3 <- H.means(dados[,-r], M, iter.max,initial.part = ultimo)
  resultado4 <- resultado3$func.objetivo
  particaofinal <- resultado3$grupos

  list(print(resultado4), print(particaofinal))

  r1<-rc


  if(resultado4 < jb) {

  jb<-resultado4

  print(jb)


    }



  }

else {
```

```
  menores<-vmenores[-c(1,2)]
  print(menores)
  rh <- which.min(menores)
  print(rh)
  fh <- menores[rh]
  print(fh)
  ultimo2 <- melit[,(rh + 2)]

  resultado5 <- H.means(dados[,-r], M, iter.max,initial.part = ultimo2)
  resultado6 <- resultado5$func.objetivo
  particaofinal2 <- resultado5$grupos
  r1<-rc

  list(print(resultado6), print(particaofinal2))
  }

int <- int + 1

}

######## Passo 5 #########

if(resultado6 < jb) {

jb<-resultado6

print(resultado6)
print(particaofinal2)

}


else {
print(jb)


}

}

###### Fim do algoritmo #######

#### Um exemplo para facilitar o uso do algoritmo é dado a seguir ###

#### Para o número de grupos igual a M=3, com k=20 perturbações, ####
#### np=20 funções objetivos, #####
```

```
### r=5, pois são 4 variáveis, logo, r= número de variáveis + 1 ###
#### e o número de iterações é igual a iter.max=15.###
#### Tal algoritmo é aplicado no banco de dados USArrests ####
### e a linha de comando para esse exemplo é dada abaixo ######

H.BaseTabu(USArrests,3,20,20,5,15)

#################
```