Universidade Federal do Ceará

Centro de Ciências

Departamento de Computação

Programa de Pós-Graduação em Ciência da Computação

Filipe Francisco Rocha Damasceno

Uma melhoria do algoritmo k-médias utilizando o estimador de James-Stein

Fortaleza

2015

FILIPE FRANCISCO ROCHA DAMASCENO

UMA MELHORIA DO ALGORITMO K-MÉDIAS UTILIZANDOO ESTIMADOR DE JAMES-STEIN

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Área de concentração: Teoria da Computação.

Orientador: Prof. Dr. João Paulo Pordeus Gomes.

Co-orientador: Prof. Dr. Carlos Eduardo Fisch de Brito.

FORTALEZA
2015

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

D162m Damasceno, Filipe Francisco Rocha.
Uma melhoria do algoritmo k-médias utilizando o estimador de James-Stein / Filipe Francisco Rocha Damasceno. – 2015.
63 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2015.
Orientação: Prof. Dr. João Paulo Pordeus Gomes.
Coorientação: Prof. Dr. Carlos Eduardo Fisch de Brito.

1. Clustering. 2. K-médias. 3. Estimador de James-Stein. I. Título.

CDD 005

FILIPE FRANCISCO ROCHA DAMASCENO

UMA MELHORIA DO ALGORITMO K-MÉDIAS UTILIZANDOO ESTIMADOR DE JAMES-STEIN

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

Área de concentração: Teoria da Computação.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

_____

Prof. Dr. João Paulo Pordeus Gomes (Orientador)
Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Carlos Eduardo Fisch de Brito (Co-orientador)
Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Amauri Holanda Souza Junior
Instituto Federal do Ceará (IFCE)

_____

Prof. Dr. José Valente de Oliveira
Universidade do Algarve (UAlg)

Este trabalho é dedicado à minha família.

Agradecimentos

Aos meus pais, Antônio Nobre e Maria de Fátima, pelo apoio incondicional durante todos os desafios que encarei na minha vida

Aos meus familiares mais próximos, minha irmã Joana D'arc e minha tia Luiza Elizabethe, pela companhia durante todo esse tempo e por todo o apoio dado

Ao meu orientador, João Paulo, pela amizade, pela paciência e pelos aconselhamentos e conversas durante o mestrado

Aos meus co-orientadores, Carlos Brito e José Valente, pelo tempo e paciência investidos em mim e pela disponibilidade para me ajudar

Aos demais professores que fizeram parte da minha caminhada, de colégio, graduação e mestrado, por todo o conhecimento repassado e por toda a paciência depositada

Aos amigos, por todos os momentos compartilhados, todas as conversas que tivemos, todas as contribuições e suporte recebidos

Resumo

A tarefa de agrupamento constitui um dos principais problemas de aprendizado de máquina. Dentre os diversos métodos propostos destaca-se o k-médias por sua simplicidade e grande aplicabilidade. É notório que o desempenho do k-médias está relacionado à estimativa dos centroides a partir dos dados e esta, usualmente, é obtida a partir da estimativa de máxima verossimilhança (EMV). Em trabalhos anteriores foi proposto um estimador denominado estimador de James-Stein (JS), sendo este capaz de, em média, superar o EMV para vetores de dados com dimensão maior que 2. Também em trabalhos anteriores foi proposta uma alteração do k-médias aplicando o estimador JS, obtendo melhoras devido à sua maior precisão em relação ao EMV. Neste trabalho propõe-se uma nova variante do algoritmo k-médias utilizando o estimador JS.

Palavras-chaves: clustering, k-médias, estimador de James-Stein.

Abstract

The clustering task constitutes one of the main machine learning problems. Among many proposed methods, k-means stands out by its simplicity and high applicability. It is notorious that k-means performance is directly related to the centroid estimation from data, which is usually obtained from the maximum likelihood estimation (MLE). In previous studies it was proposed an estimator called James-Stein (JS) estimator, being, in average, capable of overcoming MLE for vectors of data with more than 2 dimensions. Also in previous studies it was proposed a variation of k-means applying JS estimator, obtaining improvements due to its better precision when compared to MLE. In this study we propose a variation of the k-means algorithm using the JS estimator.

Key-words: clustering, k-means, James-Stein estimator.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Agrupamento (ou *Clustering*) é uma importante tarefa de Aprendizado de Máquina com aplicações em diversas áreas da Computação. É um tópico em plena evolução, visto que com o aumento da quantidade de informações geradas, especialmente incentivado pelos avanços na área da Computação, há também uma maior necessidade de se trabalhar as informações objetivando uma melhor e mais eficiente interpretação desses dados. A tarefa de agrupamento é aplicada em várias áreas da Computação, como mineração de dados (BERKHIN, 2006), segmentação de imagens (FU; MUI, 1981), reconhecimento de padrões (JAIN; DUIN; MAO, 2000), entre outras.

Muito relacionada à classificação, a tarefa de agrupamento também busca o reconhecimento de padrões em um conjunto de dados, mas sem a necessidade de um conhecimento prévio das classes dos dados. A partir de um conjunto de dados, um classificador seleciona um subconjunto desses dados e efetua o treinamento do método, utilizando a classificação real dos dados selecionados para tal, visando a classificação de outras informações que venham a ser adicionadas. Algoritmos de classificação, portanto, requerem a classificação real dos dados utilizados para treinamento, além de sua precisão depender diretamente do conjunto de treinamento selecionado.

Por sua vez, algoritmos de agrupamento utilizam todo o conjunto de dados e buscam agrupar os dados de modo que elementos de um mesmo grupo sejam similares e que elementos de grupos diferentes sejam o mais diferentes possível. Portanto, não há necessidade de treinamento ou de qualquer conhecimento prévio sobre a classificação real dos dados. Consequentemente, métodos de agrupamento são ideais para classificar dados sobre os quais não temos indicativos de uma classificação correta e sobre os quais queremos reconhecer padrões, nos ajudando a identificar grupos de dados semelhantes.

Na tarefa de agrupamento, duas abordagens são utilizadas com maior frequência: agrupamento hierárquico e agrupamento não-hierárquico. No agrupamento hierárquico, a ideia é agrupar iterativamente os dados até que alcancemos a quantidade de grupos desejada, criando, assim, uma hierarquia de grupos. Podemos fazer uma subdivisão de agrupamento hierárquico em duas outras abordagens: aglomerativa e divisiva.

No agrupamento hierárquico aglomerativo, dado um conjunto com n elementos, utilizamos uma abordagem *bottom-up*, iniciando o agrupamento com n grupos, onde cada grupo contém exatamente um elemento. A cada iteração, unimos os dois grupos mais similares, substituindo-os por um único grupo que contém os elementos dos dois grupos anteriores. Efetuamos, então, sucessivas aglomerações até obtermos a quantidade desejada de grupos.

Ilustrando essa abordagem, cada grupo será representado por um ponto, que no início do método será o único ponto pertencente a cada grupo. Para cada iteração do algoritmo, selecionamos os dois grupos que tenham os representantes mais próximos e os juntamos em um único grupo, que por sua vez será representado pelo ponto médio entre os representantes dos dois grupos anteriores. Isso posto, executamos sucessivas iterações deste tipo para obtermos um agrupamento com a quantidade desejada de grupos. Como exemplos de algoritmos que utilizam esta abordagem, podemos citar AGNES (do inglês, *Agglomerative Nesting*) (KAUFMAN; ROUSSEEUW, 1990a), *Single-linkage clustering* (GOWER; ROSS, 1969) e uma modificação deste algoritmo, SLINK (SIBSON, 1973).

Já no agrupamento hierárquico divisivo utilizamos uma abordagem *top-down*, iniciando o agrupamento com um único grupo contendo todos os elementos do conjunto de dados. A cada iteração, procuramos a melhor forma de dividir algum dos grupos existentes em dois outros e efetuamos esta divisão, substituindo o grupo dividido por dois novos grupos. Efetuamos, então, sucessivas divisões até obtermos a quantidade desejada de grupos.

Ilustrando esta abordagem, na primeira iteração dividiremos o único grupo (que contém todos os dados) em dois outros grupos. Na segunda iteração selecionaremos, dentre os dois grupos existentes, o que pode ser dividido da melhor forma, e efetuamos sua divisão, substituindo-o por dois outros grupos. Sucessivas iterações deste tipo nos levarão a um agrupamento com a quantidade desejada de grupos. Como exemplo de algoritmo que utiliza esta abordagem, podemos citar DIANA (do inglês, *Divisive Analysis*) (KAUFMAN; ROUSSEEUW, 1990b).

Podemos ilustrar a execução de ambas as abordagens utilizando a Figura 1. Na abordagem divisiva, começamos com um só grupo (representado pelo círculo, contendo todos os dados) e o dividimos em dois outros grupos (representados pelas duas maiores regiões dentro da circunferência). A divisão de um grupo resulta numa divisão da árvore de hierarquia, gerando duas novas árvores que representarão os dois novos grupos. Sucessivas divisões de grupos resultam em novas ramificações na árvore.

Já considerando uma abordagem aglomerativa, começamos com n grupos (representados pelos n pontos do conjunto de dados), onde cada grupo é associado a uma árvore, e unimos o par de grupos cujos representantes estejam mais próximos, gerando um novo grupo. Essa união resulta numa junção das árvores dos dois grupos, gerando uma nova árvore a partir das duas árvores anteriores. Sucessivas uniões de grupos, portanto, resultam em novas junções de árvores.

Em ambas as abordagens não temos a dependência em relação à classificação real dos dados e o resultado não depende de uma seleção de parte dos dados, porém temos um elevado custo computacional envolvido nas etapas de avaliação dos dois casos. A abordagem aglomerativa avalia, a cada iteração, a similaridade entre cada par de grupos

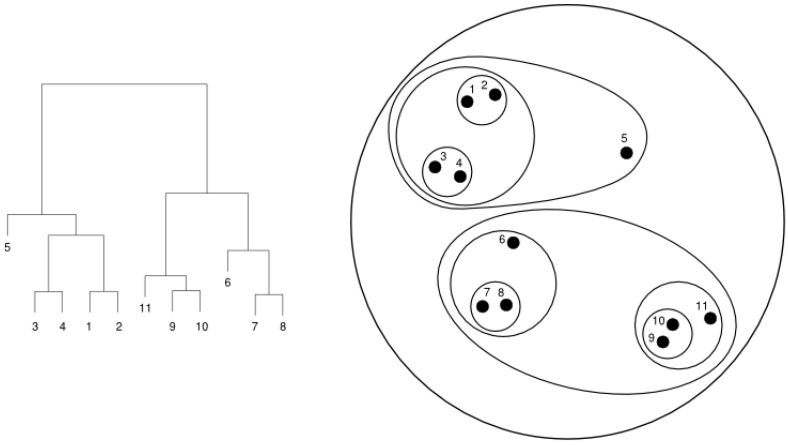

Figura 1 – Exemplo de aplicação de agrupamento hierárquico

existente para selecionar o par mais similar. Já a abordagem divisiva avalia, a cada iteração, divisões sobre todos os grupos existentes para decidir qual divisão é mais eficiente.

Tais etapas de avaliação têm um alto custo computacional à medida que utilizamos maiores conjuntos de dados. Por esse motivo, apesar dos resultados satisfatórios, abordagens hierárquicas não são muito utilizadas para tratar grandes volumes de dados. Como alternativa para os métodos hierárquicos, a abordagem particional trouxe uma nova ideia para a tarefa de agrupamento.

No agrupamento particional, buscamos separar o conjunto de dados em grupos disjuntos (ou partições) de modo que não haja grupos vazios e que cada elemento pertença a apenas uma grupo (no caso de um agrupamento *fuzzy*, elementos podem pertencer a mais de um grupo com diferentes graus de pertinência). O agrupamento particional também utiliza uma função objetivo (ou função de custo) sobre a qual o algoritmo trabalha buscando otimizar o seu valor. Por último, algoritmos particionais também se utilizam de centroides e cada grupo será representado por um único centroide, calculado a partir dos seus elementos. Com isso, em vez de procurarmos as similaridades entre pares de elementos, passamos a avaliar as similaridades entre elementos e centroides, o que acarreta uma redução de complexidade computacional do método se comparado aos métodos hierárquicos. Como exemplos de métodos de agrupamento particional, podemos citar algoritmos particionais disjuntos como k-médias (ou *k-means*) (MACQUEEN, 1967) e seus derivados (como k-medianas (JAIN; DUBES, 1988) e k-medóides (KAUFMAN; ROUSSEEUW, 1987)), e *fuzzy c-means* (RUSPINI, 1969) como algoritmo particional com sobreposição.

Além da vantagem no custo computacional se comparados aos métodos hierárquicos, algoritmos particionais também chamam a atenção pela precisão dos agrupamentos obtidos, mesmo quando consideramos algoritmos mais básicos. Por exemplo, o k-médias é um algoritmo particional simples, de baixo custo computacional e há tempos introduzido mas cujos resultados ainda são bem precisos, de modo que ele ainda é usado em

comparações com outros métodos propostos. Além disso, variantes dele (como os já citados *k-medoids* e *fuzzy c-means*, o qual podemos considerar como uma versão *fuzzy* do k-médias) continuam a ser desenvolvidas. Até mesmo algoritmos mais modernos, como o *Spectral* (SHI; MALIK, 2000), que trabalha com redução de dimensionalidade dos dados, utiliza o k-médias na sua execução.

Apesar disso, há casos em que seu resultado pode ser ruim. A ideia utilizada pelo k-médias é calcular o centroide de cada grupo a partir do estimador de máxima verossimilhança (do inglês, *Maximum Likelihood Estimator*, ou *MLE*), calculando o centroide como a média dos elementos do grupo. Além disso, o método classifica os dados de acordo com a similaridade entre dados e centroides, mas para essa classificação ele verifica apenas as distâncias deste elemento aos centroides. Com base nisso, podemos considerar que o k-médias busca apresentar cada grupo como uma distribuição gaussiana, onde cada distribuição tem média igual ao centroide do grupo, mas com todas utilizando uma mesma matriz de covariância. Assim, esse método busca estimar a média de cada distribuição para melhor agrupar os dados. Porém, há casos em que a estimativa da média dos dados não é suficientemente precisa.

Para ilustrarmos essa ideia, suponhamos um grupo gerado a partir de uma distribuição gaussiana com média e matriz de covariância desconhecidas. A precisão da estimativa da média de um grupo depende diretamente da quantidade de amostras retiradas dessa distribuição, portanto grupos com poucas amostras terão um centroide menos preciso em relação à média real da distribuição.

Dado esse contexto, uma variante do k-médias foi proposta por Hitchcock (GAO; HITCHCOCK, 2010). Essa alteração, que chamaremos de *Hitchcock JS k-means*, se baseia em alterar a estimativa de média a ser aplicada, utilizando o estimador de James-Stein (ou estimador JS). A utilização do estimador JS melhora a precisão no cálculo dos centroides dos grupos, resultando em uma maior precisão do agrupamento obtido. Essa maior precisão se justifica pela imprecisão do *MLE* quando o número de dimensões p for maior que 2, como demonstrado por Stein (STEIN, 1956).

A primeira versão do estimador de James-Stein, introduzida por Willard James e Charles Stein (JAMES; STEIN, 1961), busca, a partir de uma única amostra de uma distribuição gaussiana de média $\boldsymbol{\mu}$ e matriz de covariância $\mathbf{Q} = \sigma^2 \cdot \mathbf{I}$, ambos desconhecidos, estimar a média real $\boldsymbol{\mu}$ desta distribuição. Tal estimativa acontece através de um deslocamento da média *MLE* (no caso do *MLE*, a estimativa da média de uma única amostra é igual à própria amostra) em direção a $\mathbf{0}$. Posteriores alterações no estimador de James-Stein possibilitam sua aplicação no contexto de agrupamento a fim de melhorar a estimativa da média dos grupos.

Neste trabalho apresentamos uma nova abordagem para a utilização do estimador de James-Stein no algoritmo k-médias, tendo como base a proposta anterior de Hitchcock,

introduzindo o método *Shrinkage k-means*. No Capítulo 2 faremos uma breve revisão sobre a fundamentação teórica. No Capítulo 3 discutiremos os detalhes do método, fazendo uma comparação ao método proposto por Hitchcock. No Capítulo 4 realizamos testes comparativos entre k-médias e o método proposto, destacando alguns parâmetros de conjunto de dados que podem influenciar nessa comparação. No Capítulo 5 faremos uma breve discussão sobre o que foi apresentado e abordaremos possíveis trabalhos futuros.

2 Fundamentos Teóricos

2.1 Agrupamento

2.1.1 k-médias

O algoritmo k-médias (ou *k-means*) é um algoritmo particional disjunto inicialmente proposto por MacQueen (MACQUEEN, 1967) e cuja ideia fora introduzida anteriormente por Steinhauss (STEINHAUS, 1956). A ideia do k-médias é representar cada grupo por um centroide, que será um ponto adicional calculado a partir dos elementos do grupo, e agrupar os dados de acordo com sua similaridade em relação a esses centroides, de modo que cada elemento pertença ao grupo associado ao centroide mais similar a ele. Baseado nisso, o problema consiste em encontrar a melhor localização para os centroides.

Um ponto a ser abordado antes de tratarmos do método em si é relativo ao conceito de similaridade. Por tratarmos de pontos em um espaço p-dimensional, associaremos a ideia da similaridade entre um par de pontos à distância p-dimensional entre eles. Deste modo, quanto menor for a distância entre esses dois pontos, mais similares eles são.

A ideia dos algoritmos particionais disjuntos é agrupar os elementos do conjunto de dados em uma quantidade predefinida k de grupos de modo que não haja interseção entre grupos e que não haja grupos vazios. Estes algoritmos também utilizam centroides, que serão representantes dos grupos e serão calculados com base nos elementos do grupo.

Formalizando essa ideia, cada ponto $\mathbf{x}_j$ do conjunto de dados $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $\mathbf{x}_j \subset \mathbb{R}^p$, será associado a exatamente um grupo $\mathbf{S}_i$ em $\mathbf{S} = (\mathbf{S}_1, ..., \mathbf{S}_k)$, $\mathbf{S}_i \subseteq \mathbf{X}$, de modo que cada grupo $\mathbf{S}_i$ contenha pelo menos um elemento de $\mathbf{X}$. Cada grupo $\mathbf{S}_i$, por sua vez, será representado por um centroide $\mathbf{c}_i$ em $\mathbf{C} = (\mathbf{c}_1, ..., \mathbf{c}_k)$. No caso do k-médias, cada centroide $\mathbf{c}_i$ será calculado como a média dos elementos do grupo $\mathbf{S}_i$.

Para a avaliação do agrupamento, o k-médias utiliza uma função objetivo (ou função de custo), que será definida como a soma dos erros quadrados de cada grupo, isto é, a soma do quadrado das diferenças de cada elemento $\mathbf{x}_j$ em relação ao centroide $\mathbf{c}_i$ do grupo $\mathbf{S}_i$ ao qual ele foi associado.

Para associar elementos a grupos, o k-médias agrupa os dados de acordo com as distâncias entre elementos e centroides, de modo que um elemento $\mathbf{x}_j$ pertencerá ao grupo $\mathbf{S}_i$ cujo centroide $\mathbf{c}_i$ é o centroide mais próximo de $\mathbf{x}_j$. Expressaremos essa associação da

seguinte forma:

$$\mathbf{x}_j \in \mathbf{S}_i \iff ||\mathbf{x}_j - \mathbf{c}_i|| = \min_t ||\mathbf{x}_j - \mathbf{c}_t|| \tag{2.1}$$

Para representar a associação de um elemento $\mathbf{x}_j$ do conjunto de dados a um grupo $\mathbf{S}_i$, utilizaremos uma matriz $\mathbf{U}_{k\times n}$, chamada de matriz de partição. Por tratarmos de partições disjuntas (isto é, um elemento não pode pertencer ao mesmo tempo a dois grupos distintos), preenchemos a matriz de partição $\mathbf{U}$ com 0 e 1, de modo que $u_{ij} = 1$, com $i = 1, ..., k$ e $j = 1, ..., n$, se $\mathbf{x}_j \in \mathbf{S}_i$, e $u_{ij} = 0$ caso contrário. Portanto, utilizando a propriedade apresentada pela Equação (2.1) temos:

$$u_{ij} = \begin{cases} 1, & se\ ||\mathbf{x}_j - \mathbf{c}_i|| = \min_t ||\mathbf{x}_j - \mathbf{c}_t|| \\ 0, & caso\ contrário \end{cases} \tag{2.2}$$

Utilizando essa matriz, formalizaremos a ideia da função objetivo do k-médias, que chamaremos de Q_{km}. Teremos que essa função será definida como a soma dos erros quadrados entre cada elemento e o centroide do grupo ao qual este elemento pertence, onde o erro entre dois pontos será representado pela distância entre eles. Utilizaremos a matriz $\mathbf{U}$ para auxiliar nesse cálculo. Teremos, então, a seguinte função objetivo:

$$Q_{km} = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij} D_{ij}^2(\mathbf{x}_j, \mathbf{c}_i) \tag{2.3}$$

onde $D_{ij}(\mathbf{x}_j, \mathbf{c}_i)$ representa a distância entre o elemento $\mathbf{x}_j$ e o centroide $\mathbf{c}_i$. A função utilizada pelo k-médias para o cálculo da distância é a Euclidiana. Com base na função objetivo do k-médias, obtemos a fórmula de atualização dos centroides $\mathbf{c}_i$. Temos:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}\mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}} \tag{2.4}$$

Resumimos, então, o funcionamento do algoritmo k-médias através do Algoritmo 1.

Algoritmo 1: k-médias

1. escolher os centroides iniciais $\mathbf{c}_1, ..., \mathbf{c}_k$
2. associar cada elemento do conjunto de dados ao centroide mais próximo, preenchendo a matriz $\mathbf{U}$ como descrito na Equação (2.2) (etapa de classificação dos dados)
3. atualizar os centroides $\mathbf{c}_1, ..., \mathbf{c}_k$ de acordo com a Equação (2.4) (etapa de atualização dos centroides)
4. repetir os passos 2 e 3 até que haja convergência

O critério de convergência escolhido depende da aplicação. Alguns exemplos de critérios utilizados são: (1) a manutenção do mesmo conjunto $\mathbf{C}$ de centroides após uma iteração do algoritmo (o que equivale a não haver alterações na matriz de partição $\mathbf{U}$ após uma iteração); (2) um limite mínimo na variação da função objetivo Q_{km} entre duas iterações; e (3) um limite na quantidade de iterações do algoritmo.

Um ponto que deve ser destacado sobre o funcionamento desse algoritmo é o passo de classificação dos dados, no qual há a associação dos dados aos grupos. Pelo fato deste passo levar em conta apenas os pontos do conjunto de dados e as posições dos centroides, temos que a seleção dos centroides tem influência sobre o resultado final, podendo inclusive levar a um caso de mínimo local.

Analisando o processo iterativo do k-médias, uma seleção de centroides aleatórios ao início da execução do algoritmo nos retornará novos centroides ao final da execução, os quais são boas aproximações das médias dos grupos. Cada iteração do algoritmo, portanto, pode ser interpretada como um deslocamento dos centroides visando encontrar uma melhor posição.

Desse modo, considerando que as diferentes classes do conjunto de dados representam diferentes distribuições, posicionar os centroides iniciais próximos às médias das distribuições representa uma boa inicialização. Baseado no fato de que as iterações do algoritmo representam um deslocamento dos centroides para uma melhor posição, temos que a precisão da inicialização afeta diretamente a quantidade de iterações necessária para a convergência.

Assim, no caso de uma má escolha dos centroides iniciais temos um aumento do número de iterações necessário para se atingir a convergência. Como exemplo, podemos considerar um caso no qual os centroides escolhidos estejam muito próximos um do outro. Nesse caso, as primeiras iterações do algoritmo serão basicamente para afastar os dois centroides.

Outro ponto importante sobre a inicialização dos centroides é que uma inicialização muito ruim pode, inclusive, resultar em um caso de grupo vazio. Isto pode acontecer se inicializarmos um centroide em uma posição muito distante dos dados. Neste caso, estaríamos agrupando em $k - 1$ grupos, e não em k.

Estes pontos são trabalhados no método *k-means++* (ARTHUR; VASSILVITSKII, 2007). A escolha dos k centroides iniciais será feita selecionando, dentre os n elementos do conjunto de dados, k deles como centroides. Essa escolha é feita de uma maneira probabilística, de modo que elementos próximos a algum ponto já escolhido tenham uma menor probabilidade de serem selecionados, enquanto que elementos mais distantes de pontos já escolhidos tenham uma maior probabilidade de serem selecionados. Essa inicialização evita (ou pelo menos reduz) a necessidade de passos adicionais para separação dos

centroides. Além disso, como cada centroide é inicializado como um ponto do conjunto de dados, temos que cada grupo conterá pelo menos um elemento, garantindo, assim, que não haverão grupos vazios.

Outro ponto importante a se ressaltar é que a qualidade da classificação depende tanto da distância entre grupos, quanto de quão unidos são os elementos de cada grupo. Quanto mais distantes os grupos, melhor podemos separá-los e mais precisa será a classificação. Do mesmo modo, quanto mais unidos estiverem os elementos de cada grupo, mais fácil será para diferenciá-los. Esses fatos são ilustrados através da Figura 2. De modo contrário, se temos grupos muito próximos ou cujos elementos não sejam bem unidos, pode haver um comprometimento da qualidade da classificação devido à sobreposição entre os grupos, como podemos ver na Figura 3.

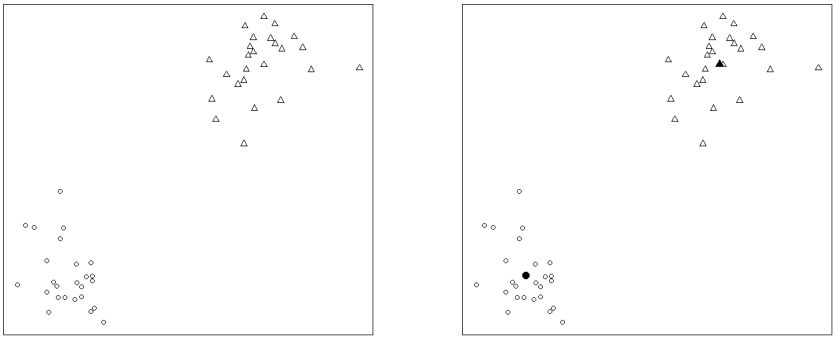

Figura 2 – Exemplo de conjunto de dados com grupos bem separados, comparando a classificação real dos dados (esq.) com a classificação obtida através de uma execução completa do k-médias (dir.), destacando os centroides obtidos

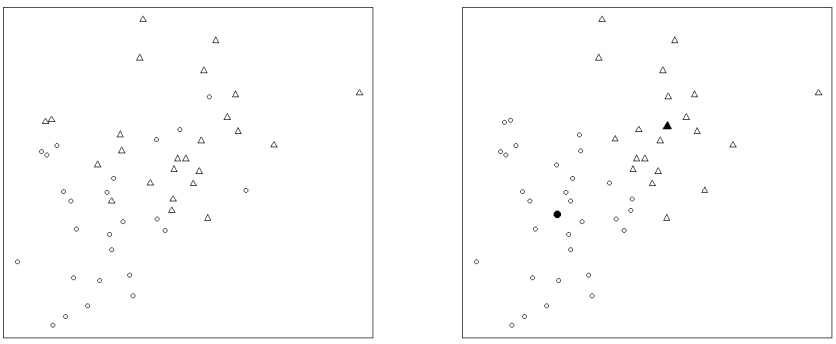

Figura 3 – Exemplo de conjunto de dados com sobreposição de grupos, comparando a classificação real dos dados (esq.) com a classificação obtida através de uma execução completa do k-médias (dir.), destacando os centroides obtidos

O k-médias é um algoritmo amplamente utilizado na área de agrupamento, prin-

cipalmente pela sua simplicidade e rápida convergência. Ele garante a convergência para um ótimo em uma quantidade finita de passos (SELIM; ISMAIL, 1984). Apesar disso, ele não tem convergência garantida para um mínimo global devido à dependência em relação à inicialização dos centroides. Portanto, para encontrar um melhor agrupamento, executa-se o algoritmo repetidas vezes, com diferentes inicializações dos centroides, e selecionamos o agrupamento que minimize a função objetivo Q_{km}. Note que a ideia de várias execuções do k-médias é viável devido ao baixo custo computacional do método.

2.1.2 k-medianas

O algoritmo k-medianas (ou *k-medians*) é um algoritmo particional disjunto derivado do k-médias. Seu funcionamento é semelhante ao k-médias, exceto pelo fato de que em vez de calcular cada centroide como a média dos elementos do grupo, calcularemos o centroide de cada grupo como a mediana dos seus elementos. Em um caso com dados p-dimensionais, a mediana será calculada a partir da mediana de cada uma das dimensões.

Por derivar do k-médias, o k-medianas tem um funcionamento semelhante, buscando, a partir de um conjunto de dados $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$, encontrar k centroides $\mathbf{c}_i$, $i = 1, ..., k$ que minimizem a soma das distâncias de cada $\mathbf{x}_j$, $j = 1, ..., n$, ao centroide mais próximo.

A classificação dos dados ocorre da mesma forma que no k-médias, classificando os dados de acordo com a proximidade em relação aos centroides:

$$\mathbf{x}_j \in \mathbf{S}_i \iff ||\mathbf{x}_j - \mathbf{c}_i|| = \min_t ||\mathbf{x}_j - \mathbf{c}_t|| \tag{2.5}$$

A principal diferença reside no cálculo dos centroides. Na sua etapa de atualização dos centroides, o k-medianas os calcula como a mediana dos elementos do grupo, a qual será calculada a partir da mediana de cada uma das p dimensões. Note que o centroide, assim como no k-médias, não é considerado como um elemento do grupo, mas sim como um ponto representante do grupo.

Podemos resumir o funcionamento do k-medianas através do Algoritmo 2.

Algoritmo 2: k-medianas

1 escolher os centroides iniciais $\mathbf{c}_1, ..., \mathbf{c}_k$
2 associar cada elemento do conjunto de dados ao centroide mais próximo, de acordo com a Equação (2.5) (etapa de classificação dos dados)
3 atualizar os centroides $\mathbf{c}_1, ..., \mathbf{c}_k$ como a mediana de cada grupo $\mathbf{S}_i$, $i = 1, ..., k$ (etapa de atualização dos centroides)
4 repetir os passos 2 e 3 até que haja convergência

O k-medianas tem uma maior tolerância a *outliers* se comparado ao k-médias, visto que a seleção da mediana dos elementos em vez da média pode desconsiderar valores muito maiores (ou muito menores) que os demais. Por exemplo, se considerarmos um caso unidimensional de $\mathbf{X} = (1, 2, 3, 4, 100000)$, enquanto que a média simples de $\mathbf{X}$ é igual a 20002, a mediana de $\mathbf{X}$ é igual a 3. Este conjunto dá a ideia de que 100000 é um *outlier*, demonstrando uma maior precisão da mediana se comparada à média.

Um erro comum quando se trabalha com a mediana é confundir seu conceito com o de medóide. Ambos buscam minimizar o erro em relação aos dados, porém enquanto que a mediana é um ponto adicional calculado a partir dos dados que temos e não necessariamente igual a algum dos dados, o medóide é um ponto adicional que será igual a algum dos dados. No caso do medóide, selecionamos o ponto que minimize o erro em relação a todos os pontos, enquanto que no caso da mediana selecionamos medianas de atributos para minimizar o erro.

Apesar de mediana e medóide terem exemplos de igualdade no caso unidimensional, como acontece no exemplo anterior (onde ambos mediana e medóide são iguais a 3), podemos exemplificar a diferença no caso bidimensional. Considerando um conjunto $\mathbf{X} = ((1, 1), (2, 2), (3, 3))$, temos ambos mediana e medóide iguais a $(2, 2)$. Porém, se considerarmos um novo conjunto $\mathbf{X}' = ((1, 2), (2, 3), (3, 1))$, teremos uma mediana igual a $(2, 2)$, ponto que não pertence a $\mathbf{X}'$ e que, portanto, não é igual ao medóide.

2.2 Estimador de James-Stein

O estimador de James-Stein é um estimador de média cujo objetivo é, a partir de uma única amostra de uma distribuição gaussiana desconhecida, estimar a média real da distribuição. Para esse problema, foi provado que o *MLE* (*Maximum Likelihood Estimator*, do inglês, estimador de máxima verossimilhança) não era ótimo quando o número de dimensões p fosse maior que 2, e que o estimador de James-Stein alcançava um erro médio menor que o *MLE*.

2.2.1 O problema da estimativa

Considere que temos apenas um ponto p-dimensional $\mathbf{x}$, que é uma amostra (ou observação) de uma distribuição gaussiana de média $\boldsymbol{\theta}$ e matriz de covariância $\sigma^2 \cdot \mathbf{I}_p$, ambas desconhecidas. O problema da estimativa se resume em fazer uma estimativa $\hat{\boldsymbol{\theta}}$ da média $\boldsymbol{\theta}$ de uma distribuição gaussiana $\mathcal{N}(\boldsymbol{\theta}, \sigma^2 \cdot \mathbf{I}_p)$ baseado nessa única amostra $\mathbf{x}$ dessa distribuição (com $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2 \cdot \mathbf{I}_p)$). Inicialmente, vamos considerar $\mathbf{x}$ um vetor de medidas não-correlacionadas e de igual variância.

Pelo fato de nossa estimativa $\hat{\boldsymbol{\theta}}$ depender apenas da amostra $\mathbf{x}$, podemos dizer que ela é uma função dessa observação, ou $\hat{\boldsymbol{\theta}} = \delta(\mathbf{x})$, onde onde δ é uma regra de decisão (que

representará a maneira pela qual obtivemos nossa estimativa).

Para a avaliação de uma estimativa, faremos uma comparação entre $\hat{\boldsymbol{\theta}}$ e a média real $\boldsymbol{\theta}$ da distribuição. Para isso, utilizaremos uma função de risco R que calcule o erro esperado entre $\boldsymbol{\theta}$ e $\hat{\boldsymbol{\theta}}$. Como função de erro, utilizaremos o erro quadrático. Como consequência, teremos que R será igual ao *MSE* (*Mean-Squared Error*, do inglês, erro médio quadrático). Por fim, podemos definir nossa função de risco R da seguinte forma:

$$R(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \mathbb{E}[L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})] = MSE(\hat{\boldsymbol{\theta}}) = \frac{1}{p}\sum_{i=1}^{p}(\hat{\theta}_i - \theta_i)^2$$

2.2.2 *MLE* e o estimador de James-Stein

Vc deve escolher entre usar MLE ou EMV.

Comumente se utilizava o *MLE* para estimar a média $\boldsymbol{\theta}$. Se baseando no fato de que $\mathbf{x}$ segue uma distribuição gaussiana e, portanto, a média dos ruídos é nula, o *MLE* utiliza a própria observação $\mathbf{x}$ como estimativa da média $\boldsymbol{\theta}$, ou seja:

$$\hat{\boldsymbol{\theta}}_{MLE} = \delta_{MLE}(\mathbf{x}) = \mathbf{x}$$

Acreditava-se que esse era o melhor estimador para esse caso, e seu erro, por aumentar à medida que a dimensão era aumentada (como demonstrado através do erro acima), era aceitável. Porém, Stein demonstrou que o *MLE* era inadmissível quando a dimensão p dos dados era maior do que 2 (STEIN, 1956). Essa prova de inadmissibilidade demonstrou que o *MLE* não é admissível quando $p > 2$, o que significa que há algum estimador que domina o *MLE* neste caso.

Posteriormente, Willard James e Charles Stein propuseram um novo estimador (JAMES; STEIN, 1961). Chamado de estimador de James-Stein (ou estimador JS), ele é definido da seguinte forma:

$$\hat{\boldsymbol{\theta}}_{JS} = \delta_{JS}(\mathbf{x}) = \left(1 - \frac{\sigma^2(p-2)}{\mathbf{x}^T\mathbf{x}}\right) \cdot \mathbf{x}$$

Note que se $\sigma^2(p-2) < \mathbf{x}^T\mathbf{x}$ então o coeficiente $\left(1 - \frac{\sigma^2(p-2)}{\mathbf{x}^T\mathbf{x}}\right)$ empurra $\mathbf{x}$ em direção a $\mathbf{0}$, agindo como um coeficiente de *shrinkage* (ou encolhimento).

Além de propor esse novo estimador, eles mostraram que o estimador de James-Stein dominava o *MLE* nessas condições. Em outras palavras, seu estimador tinha uma performance média melhor que o *MLE* quando $p > 2$, ou seja:

$$MSE(\hat{\boldsymbol{\theta}}_{JS}) < MSE(\hat{\boldsymbol{\theta}}_{MLE})$$

2.2.3 Análise dos estimadores

seção praticamente refeita

Para a análise do desempenho dos estimadores, faremos a comparação entre os valores da função de risco para o *MLE* e para o estimador JS. Ao utilizarmos o *MLE* para a estimativa, temos o seguinte risco:

$$R(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_{MLE}) = \mathbb{E}\left[||\hat{\boldsymbol{\theta}}_{MLE} - \boldsymbol{\theta}||^2\right] = p\sigma^2 \tag{2.6}$$

Para o estimador JS, temos o seguinte risco:

$$R(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_{JS}) = p\sigma^2 - (p-2)\sigma^2 \cdot \exp\left[-\frac{\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\sigma^2}\right] \cdot \sum_{n=0}^{\infty} \frac{(\boldsymbol{\theta}^T\boldsymbol{\theta})^n}{(2\sigma^2)^n(p-2+2n)n!} \tag{2.7}$$

A comparação entre os estimadores pode ser feita utilizando as equações (2.6) e (2.7). Para isso, basta-nos analisar a diferença entre os riscos, apontada pelo termo negativo presente na equação (2.7), e o efeito da variação dos parâmetros sobre ele.

Inicialmente, a análise do termo nos mostra que quando consideramos o caso de $p > 2$, o termo não assume valores negativos. Como consequência disto, temos que o risco do estimador JS tem um mínimo de $p\sigma^2$. Isso nos permite afirmar que no pior dos casos o estimador JS tem uma precisão tão boa quanto o *MLE*.

Ainda relacionado à quantidade p de dimensões, temos que o valor do termo aumenta à medida que aumentamos o valor de p. Como consequência, temos que à medida que aumentamos a quantidade de dimensões dos dados, maior se torna a diferença entre os riscos dos estimadores, com uma maior precisão obtida pelo estimador JS.

Outro ponto a ser ressaltado sobre a função de risco diz respeito à distância de $\boldsymbol{\theta}$ a $\mathbf{0}$. Pela análise do termo da equação (2.7), verificamos que seu valor aumenta à medida que $||\boldsymbol{\theta}||^2 \to \mathbf{0}$, além de que à medida que $||\boldsymbol{\theta}||^2 \to \infty$ o valor da função de risco para o estimador JS se aproxima do valor para o *MLE*.

Por último, dada uma amostra $\mathbf{x}$ de uma distribuição (com $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \mathbf{Q})$) podemos utilizar os valores de $||\mathbf{x}||^2$ e $||\boldsymbol{\theta}||^2$ para analisar o encolhimento efetuado pelo estimador JS. Considerando uma distribuição com média relativamente distante de $\mathbf{0}$, se $||\mathbf{x}||^2 < ||\boldsymbol{\theta}||^2$ então o encolhimento em direção a $\mathbf{0}$ fará com que a amostra seja encolhida de modo a afastá-la da média da distribuição, assim aumentando a diferença entre $\mathbf{x}$ e $\boldsymbol{\theta}$ e, portanto, obtendo uma aproximação pior do que a amostra utilizada.

Por esse motivo, deve-se analisar a probabilidade de uma amostra dessa distribuição estar mais distante de $\mathbf{0}$ do que a média. Para isso, calcularemos o valor esperado da

norma quadrática de $\mathbf{x}$ e utilizaremos a diferença entre $\mathbb{E}[||\mathbf{x}||^2]$ e $||\boldsymbol{\theta}||^2$ como indicativo da probabilidade de uma amostra ter norma quadrática maior que a média.

Sabendo que $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \mathbf{Q})$, com $\mathbf{x}$ sendo um vetor p-dimensional, e considerando $\boldsymbol{\Lambda}$ uma matriz simétrica de dimensões $p \times p$, temos a seguinte igualdade (KOCH, 2007):

$$\mathbb{E}[\mathbf{x}^T \boldsymbol{\Lambda} \mathbf{x}] = \boldsymbol{\theta}^T \boldsymbol{\Lambda} \boldsymbol{\theta} + tr(\boldsymbol{\Lambda}\mathbf{Q}) \tag{2.8}$$

onde $tr(\boldsymbol{\Sigma})$ é o traço de uma matriz $\boldsymbol{\Sigma}$. Num caso especial, quando $\boldsymbol{\Lambda} = \mathbf{I}_p$ (onde $\mathbf{I}_p$ é a matriz identidade p-dimensional), podemos reescrever a equação (2.8) da seguinte forma:

$$\mathbb{E}[\mathbf{x}^T \mathbf{x}] = \boldsymbol{\theta}^T \boldsymbol{\theta} + tr(\mathbf{Q}) \tag{2.9}$$

o que equivale a:

$$\mathbb{E}[||\mathbf{x}||^2] = ||\boldsymbol{\theta}||^2 + tr(\mathbf{Q}) \tag{2.10}$$

Pela equação acima, podemos afirmar que o valor esperado da norma quadrática de $\mathbf{x}$ é maior que a norma quadrática de $\boldsymbol{\theta}$. Isso se deve ao fato de que $tr(\mathbf{Q})$ é a soma das variâncias de cada dimensão (presentes na diagonal principal da matriz de covariância), que não assumem valores negativos.

Também pela equação (2.10) podemos analisar outros fatores relacionados ao valor esperado de $||\mathbf{x}||^2$. Primeiro, o valor esperado de $||\mathbf{x}||^2$ aumenta à medida que aumentamos as variâncias presentes na matriz $\mathbf{Q}$. Segundo, à medida que aumentamos o número p de dimensões, também aumentamos o valor esperado de $||\mathbf{x}||^2$, visto que quanto maior o valor de p, maior será, em média, a diferença entre $||\mathbf{x}||^2$ e $||\boldsymbol{\theta}||^2$.

Por fim, temos que à medida que aproximamos $\boldsymbol{\theta}$ de $\mathbf{0}$ (ou distanciamos $\mathbf{x}$ de $\mathbf{0}$) aumentamos também a probabilidade de que nossa amostra $\mathbf{x}$ tenha uma norma quadrática $||\mathbf{x}||^2$ maior que $||\boldsymbol{\theta}||^2$. Em um caso extremo, se $\boldsymbol{\theta} = \mathbf{0}$ temos $||\boldsymbol{\theta}||^2 = 0$ e, portanto, a probabilidade de selecionarmos uma amostra com norma quadrática menor que $||\boldsymbol{\theta}||^2$ é igual a zero. Já ao distanciarmos $\boldsymbol{\theta}$ de $\mathbf{0}$, essa probabilidade aumenta, mas como visto, se mantém menor que a probabilidade de selecionar uma amostra com norma quadrática maior que $||\boldsymbol{\theta}||^2$. Este ponto, portanto, concorda com a análise anteriormente feita sobre os efeitos do valor de $||\boldsymbol{\theta}||^2$ na equação (2.7).

2.2.4 Derivações do estimador JS

Uma grande limitação do estimador de James-Stein é a restrição de que a variância é fixa. Bock introduziu uma derivação do estimador para que os atributos de $\mathbf{x}$ possam ter correlação e diferentes variâncias (BOCK, 1975). Para isso, consideraremos $\mathbf{x}$ uma

observação que segue uma distribuição gaussiana p-dimensional multivariada com média $\boldsymbol{\theta}$ e matriz de covariância $\mathbf{Q}$ de dimensões $p \times p$, com $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \mathbf{Q})$. Teremos, então, o seguinte estimador:

$$\hat{\boldsymbol{\theta}}_{JS}^{(1)} = \left(1 - \frac{\hat{p} - 2}{\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x}}\right) \cdot \mathbf{x} \tag{2.11}$$

Onde $\hat{p}$ é a dimensão efetiva de $\mathbf{Q}$, definida pela divisão entre o traço de $\mathbf{Q}$ sobre o maior autovalor dessa matriz, ou:

$$\hat{p} = \frac{tr(\mathbf{Q})}{\lambda_{max}(\mathbf{Q})}$$

Outro ponto importante sobre o coeficiente desse estimador é o fato de que ele pode se tornar negativo caso a fração do seu coeficiente assuma valor maior que 1. Neste caso, em vez de encolher a média *MLE* em direção a 0, o encolhimento efetuado pelo estimador JS a empurrará na direção oposta a 0. Para tratar esse problema, utilizaremos a ideia de y^+ como a parte positiva de um valor y. Definiremos y^+ como:

$$y^+ = \begin{cases} y, & y \geq 0 \\ 0, & y < 0 \end{cases}$$

Baseado nisso, aplicaremos essa definição de parte positiva no estimador (2.11) para garantir que seu coeficiente não assuma valores negativos. O resultado obtido é um novo estimador, definido da seguinte forma (BARANCHIK, 1964):

$$\hat{\boldsymbol{\theta}}_{JS}^{(2)} = \left(1 - \frac{\hat{p} - 2}{\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x}}\right)^+ \cdot \mathbf{x} \tag{2.12}$$

Finalmente, suponha que temos uma aproximação $\bar{\boldsymbol{\theta}}$ da média $\boldsymbol{\theta}$. Podemos, então, utilizar o estimador para direcionar nossa estimação $\hat{\boldsymbol{\theta}}_{JS}^{(2)}$ em direção a $\bar{\boldsymbol{\theta}}$. Para isso, basta estimarmos $\hat{\boldsymbol{\theta}}_{JS}^{(2)} - \bar{\boldsymbol{\theta}}$ a partir de $\mathbf{x} - \bar{\boldsymbol{\theta}}$, e adicionar $\bar{\boldsymbol{\theta}}$ como um deslocamento. Teremos, assim, nosso estimador final definido da seguinte forma:

$$\hat{\boldsymbol{\theta}}_{JS} = \bar{\boldsymbol{\theta}} + \left(1 - \frac{\hat{p} - 2}{\left(\mathbf{x} - \bar{\boldsymbol{\theta}}\right)^T \mathbf{Q}^{-1} \left(\mathbf{x} - \bar{\boldsymbol{\theta}}\right)}\right)^+ \left(\mathbf{x} - \bar{\boldsymbol{\theta}}\right) \tag{2.13}$$

A ideia por trás desse novo estimador é utilizar a precisão obtida pelo estimador básico, mas com . Temos que o estimador JS clássico consegue bons resultados na estimativa de $\boldsymbol{\theta}$ ao encolher $\mathbf{x}$ em direção a $\mathbf{0}$. Partindo disso, se tivermos uma aproximação $\bar{\boldsymbol{\theta}}$ da média, podemos utilizar essa informação para fazer esse *shrinkage*, aproximando assim

nossa estimativa $\hat{\boldsymbol{\theta}}$ da média $\boldsymbol{\theta}$, de modo que quanto melhor for nossa aproximação $\bar{\boldsymbol{\theta}}$, melhor será o resultado obtido pelo estimador.

Esta última forma do estimador de James-Stein representa um caso geral do estimador inicial, se aproveitando dos pontos apresentados na análise feita anteriormente mas representando um caso mais geral. Por exemplo, se $\bar{\boldsymbol{\theta}} = \mathbf{0}$

2.2.5 Exemplo de aplicação

Para ilustrar o funcionamento do estimador JS e para uma comparação dos resultados com o *MLE*, utilizaremos os dados sobre as rebatidas de 18 jogadores da liga de baseball norte-americana na temporada de 1970 (EFRON; MORRIS, 1975).

Para cada atleta, temos uma média de acertos nas primeiras 45 tentativas de rebatidas no campeonato. Como temos uma mesma quantidade de tentativas para cada jogador, é justa a comparação com base na média de acertos, calculada como a quantidade de acertos dividida pela quantidade de tentativas. Além disso, também temos a média final de cada jogador na temporada. Estas médias são apresentadas na Tabela 1.

Batedor	Média parcial ($\mathbf{x}$)	Média da temporada ($\mathbf{y}$)
Clemente	0.400	0.346
Robinson	0.378	0.298
Howard	0.356	0.276
Johnstone	0.333	0.222
Berry	0.311	0.273
Spencer	0.311	0.270
Kessinger	0.289	0.263
Alvarado	0.267	0.210
Santo	0.244	0.269
Swoboda	0.244	0.230
Unser	0.222	0.264
Williams	0.222	0.256
Scott	0.222	0.303
Petrocelli	0.222	0.264
Rodriguez	0.222	0.226
Campaneris	0.200	0.285
Munson	0.178	0.316
Alvis	0.156	0.200

Tabela 1 – Dados de rebatidas para 18 jogadores, com suas médias parciais (em um mesmo momento da temporada) e suas médias ao final da temporada

Nosso objetivo é, baseado na média parcial de acertos dos jogadores, estimar a média final de acertos de cada um dos jogadores ao final da temporada. Para comparar os resultados, obteremos as estimativas $\hat{\boldsymbol{\theta}}_{MLE}$ e $\hat{\boldsymbol{\theta}}_{JS}$ e faremos uma comparação dos erros em relação às médias finais da Tabela 1.

Para isso, seja $\mathbf{x}$ o vetor de 18 dimensões com as médias parciais de rebatidas de cada atleta e seja σ o desvio padrão das médias parciais de acertos. Primeiramente, estimaremos a média usando o *MLE*. Por definição, teremos que a estimativa $\hat{\boldsymbol{\theta}}_{MLE}$ das médias finais dos 18 jogadores será igual às médias parciais, ou seja:

$$\hat{\boldsymbol{\theta}}_{MLE} = \mathbf{x}$$

Para o cálculo da estimativa pelo estimador de James-Stein, partiremos do fato de que temos um desvio padrão fixo. Com isso, obtemos a aproximação utilizando o estimador de James-Stein com a aplicação dos conceitos de parte positiva e de que temos uma aproximação $\bar{\boldsymbol{\theta}}$ da média. Inicialmente, utilizaremos a média das 18 médias parciais como aproximação.

Utilizando esses dois conceitos sobre o estimador, estimaremos as médias finais dos jogadores utilizando o estimador JS da seguinte forma:

$$\hat{\boldsymbol{\theta}}_{JS} = \overline{\boldsymbol{\theta}} + \left(1 - \frac{p-2}{||\mathbf{x}||^2}\right)^{+} \left(\mathbf{x} - \overline{\boldsymbol{\theta}}\right)$$

onde p será igual à dimensão dos dados, nesse caso sendo a quantidade de jogadores, igual a 18.

Para a avaliação dos resultados obtidos, faremos uma soma dos erros quadráticos para comparar as estimativas de médias às médias reais dos jogadores ao final da temporada. Para isso, considere $\mathbf{y}$ o vetor com as médias dos 18 jogadores ao final da temporada. Calcularemos os erros da seguinte forma:

$$E = \sum_{i=1}^{p} \left(\hat{\theta}_i - y_i\right)^2$$

As estimativas obtidas estão listadas na Tabela 2, assim como as médias ao final da temporada. Obtivemos as somas dos erros quadráticos iguais a $E_{MLE} = 0.075374$ e $E_{JS} = 0.02178568$, mostrando que obtivemos uma melhor aproximação das médias da temporada utilizando o estimador de James-Stein se comparado ao *MLE*.

Um ponto importante a ser ressaltado sobre o funcionamento do estimador JS é que sua melhora no resultado em relação ao *MLE* é garantida não quando vemos os casos individualmente, mas sim na média dos resultados. No exemplo dado, poderíamos apontar alguns jogadores (como Swoboda e Rodriguez) sobre os quais o *MLE* teve maior precisão em relação à média final. Apesar disso, considerando a média dessas diferenças, o estimador JS tem uma vantagem sobre o *MLE*.

Outro ponto importante a se citar é que a aproximação $\bar{\boldsymbol{\theta}}$ escolhida é a média das médias parciais dos 18 jogadores. Poderiamos ter utilizado outros valores, como

Batedor	$\hat{\theta}_{MLE}$	$\hat{\theta}_{JS}$	Média da temporada ($\mathbf{y}$)
Clemente	0.400	0.280	0.346
Robinson	0.378	0.277	0.298
Howard	0.356	0.275	0.276
Johnstone	0.333	0.272	0.222
Berry	0.311	0.270	0.273
Spencer	0.311	0.270	0.270
Kessinger	0.289	0.268	0.263
Alvarado	0.267	0.265	0.210
Santo	0.244	0.263	0.269
Swoboda	0.244	0.263	0.230
Unser	0.222	0.260	0.264
Williams	0.222	0.260	0.256
Scott	0.222	0.260	0.303
Petrocelli	0.222	0.260	0.264
Rodriguez	0.222	0.260	0.226
Campaneris	0.200	0.258	0.285
Munson	0.178	0.255	0.316
Alvis	0.156	0.253	0.200

Tabela 2 – Aproximações obtidas utilizando as estimativas *MLE* e James-Stein

por exemplo, utilizar $\mathbf{0}$ ou $\mathbf{1}$ como aproximação. Com esses valores podemos, inclusive, reforçar o fato de que o estimador JS domina o *MLE*.

Ao utilizarmos $\mathbf{0}$ e $\mathbf{1}$ como aproximações da média e fazer o *shrinkage* em direção a esses pontos, obtemos as somas dos erros quadráticos iguais a $E_{JS}^{(0)} = 0.07192643$ quando encolhemos em direção a $\mathbf{0}$ e $E_{JS}^{(1)} = 0.07498864$ quando encolhemos em direção a $\mathbf{1}$, contra a mesma soma $E_{MLE} = 0.075374$ para o *MLE*. Isso nos mostra que mesmo com aproximações não tão precisas nós ainda obtemos uma melhor estimativa ao utilizar o estimador JS.

Num último exemplo, caso utilizemos a média final $\mathbf{y}$ como aproximação $\bar{\boldsymbol{\theta}}$ da média para o estimador JS, temos um erro $E_{JS} = 0.00005475519$. Esse exemplo mostra que quanto melhor for a aproximação $\bar{\boldsymbol{\theta}}$ utilizada (isto é, quanto mais precisa em relação à média real), melhor será o resultado obtido.

3 Metodologia proposta

Nesse trabalho, propomos uma alteração ao algoritmo k-médias. Nossa proposta se baseia no método apresentado por Hitchcock (GAO; HITCHCOCK, 2010). Neste trabalho, a ideia da utilização do estimador James-Stein para estimativa dos centroides foi primeiramente apresentada. Nosso trabalho também utiliza o estimator de James-Stein para este mesmo fim, porém apresentando uma nova forma de cálculo para o fator de encolhimento (ou *shrinkage*), além de um procedimento para escolha da aproximação, que definirá a direção do encolhimento.

Desta forma, antes da apresentação do método proposto será aprentado o trabalho de Hitchcock. Antes disso, porém, devemos introduzir o conceito do *Rand Index*, um índice de qualidade de agrupamento utilizado em ambos os métodos.

3.1 Rand Index

O *Rand Index* (RAND, 1971) é um índice de qualidade proposto por William Rand que tem como objetivo avaliar a similaridade entre duas configurações. Para isso, consideraremos que a configuração resultante de um agrupamento é um vetor de tamanho n (onde n é o número de elementos do conjunto de dados) preenchido por valores em $1, ..., k$ (onde k é o número de grupos), de modo que o n-ésimo elemento do conjunto de dados está associado ao grupo armazenado na n-ésima posição do vetor.

O funcionamento desse índice parte da comparação entre pares de elementos, analisando os grupos aos quais eles foram associados em cada configuração. Vamos considerar duas configurações $\mathbf{x}$ e $\mathbf{y}$. Seja n_{00} o número de pares de elementos que estão em grupos diferentes tanto em $\mathbf{x}$ quanto em $\mathbf{y}$. Seja n_{01} o número de pares de elementos que estão em grupos diferentes em $\mathbf{x}$ e em grupos iguais em $\mathbf{y}$. Seja n_{10} o número de pares de elementos que estão em grupos iguais em $\mathbf{x}$ e em grupos diferentes em $\mathbf{y}$. Seja n_{11} o número de pares de elementos que estão em grupos iguais tanto em $\mathbf{x}$ quanto em $\mathbf{y}$. O índice RI que compara as configurações $\mathbf{x}$ e $\mathbf{y}$ será calculado da seguinte forma:

$$RI(\mathbf{x}, \mathbf{y}) = \frac{n_{00} + n_{11}}{n_{00} + n_{01} + n_{10} + n_{11}} = \frac{n_{00} + n_{11}}{\binom{n}{2}}$$

RI tem um valor máximo igual a 1, e valores maiores de RI indicam uma boa similaridade. Para um melhor entendimento desse índice, utilizaremos um exemplo simples. Considere duas configurações $\mathbf{x} = (1, 1, 1, 2, 2)$ e $\mathbf{y} = (1, 1, 2, 2, 2)$. Teremos que os pares de elementos 1 e 4, 1 e 5, 2 e 4, e 2 e 5, estão em grupos diferentes tanto em $\mathbf{x}$

quanto em $\mathbf{y}$, logo $n_{00} = 4$. Teremos também que os pares de elementos 1 e 2, e 4 e 5, estão em grupos iguais tanto em $\mathbf{x}$ quanto em $\mathbf{y}$, logo $n_{11} = 2$. Como $\binom{n}{2} = 10$, teremos que $RI(\mathbf{x}, \mathbf{y}) = 0.6$.

Pelo exemplo acima, podemos também falar sobre uma comparação básica das configurações (isto é, uma comparação entre duas configurações, elemento a elemento, verificando a quantidade de igualdades entre as configurações de cada um dos n pontos). No exemplo acima, 4 dos 5 elementos pertencem ao mesmo grupo nas duas configurações, logo a precisão retornada pela classificação básica seria 0.8. Com essa base, podemos apresentar algumas vantagens do *Rand Index* sobre esse tipo de comparação.

Uma das vantagens vem à tona quando temos duas configurações como $\mathbf{x} = (1, 1, 1, 2, 2)$ e $\mathbf{y} = (2, 2, 2, 1, 1)$. Visivelmente, podemos dizer que os agrupamentos representados por essas duas configurações são iguais, já que os grupos estão igualmente separados (há dois grupos $\mathbf{S}_1 = (1, 2, 3)$ e $\mathbf{S}_2 = (4, 5)$ em ambas as configurações). Porém, se fizermos uma comparação básica entre $\mathbf{x}$ e $\mathbf{y}$, não teremos nenhuma igualdade nessa classificação. Já com o uso do *Rand Index*, teremos que $n_{00} = 10$ e $n_{11} = 0$, e teremos $RI(\mathbf{x}, \mathbf{y}) = 1$, indicando que as duas configurações representam uma igual separação entre os grupos ou, em outras palavras, os dois agrupamentos são iguais.

Uma outra vantagem do *Rand Index* é o fato de ele analisar quão bem elementos de grupos diferentes são separados. Isso é feito ao observamos os valores n_{01} e n_{10}, que indicam que em uma das configurações há um par de elementos no mesmo grupo, e na outra configuração esse mesmo par de elementos está em grupos diferentes. Neste caso, podemos considerar que eles não devem estar bem separados, o que é considerado um ponto negativo.

Este fato é confirmado ao analisarmos a equação do *Rand Index*. Temos que o valor da função diminui à medida que aumentamos os valores de n_{01} e n_{10}, o que é um indicativo de que estes são valores indesejados numa comparação entre configurações. Também temos que o valor da função aumenta à medida que aumentamos os valores de n_{00} e n_{11}, o que é um indicativo de que há uma boa separação entre os elementos em ambas as configurações, assim sendo considerado um resultado desejado neste tipo de comparação.

Por fim, podemos utilizar o *Rand Index* como um índice de qualidade de uma configuração. Para isto, basta-nos fazer a comparação entre a configuração resultante de um algoritmo de agrupamento e a configuração real dos dados. Então, o *Rand Index* resultante seria uma espécie de grau de precisão do agrupamento obtido. Além deste uso, também utilizaremos o *Rand Index* como critério de convergência, utilizando-o para analisar se houve alteração entre duas configurações sucessivas. Se o índice entre elas for igual a 1, as duas configurações são similares, então não houve alteração.

3.2 Hitchcock JS k-means

O método introduzido por Hitchcock (GAO; HITCHCOCK, 2010), que chamaremos de *Hitchcock JS k-means*, utiliza o estimador de James-Stein sobre os centroides calculados por cada iteração do k-médias, com o objetivo de calcular novos pontos e utilizá-los como centroides. A ideia do método é utilizar o estimador JS para deslocar os centroides obtidos pelo k-médias, obtendo melhores estimativas das médias de cada grupo.

Suponha que cada grupo de dados é formado a partir de uma certa distribuição. Intuitivamente, ao calcularmos o centroide de um grupo no k-médias como a média dos seus elementos, estamos calculando uma média *MLE* desse grupo, o que representa uma estimativa da média da distribuição deste grupo. Como mostramos anteriormente, o estimador JS domina o *MLE* quando $p > 2$ e é, no pior dos casos, tão bom quanto o *MLE*. Assim, a proposta do autor é substituir a estimativa *MLE* do k-médias pelo estimador de James-Stein, obtendo melhores aproximações das médias das distribuições e, por consequência, obtendo um melhor agrupamento resultante.

Para essa aplicação, Hitchcock utiliza o estimador JS para direcionar a média *MLE* para uma aproximação da média. A aproximação de média escolhida pelo autor foi a média global dos dados (média de todos os pontos do conjunto de dados), sendo utilizada no encolhimento dos centroides de todos os grupos. Assim, o autor comprime cada um dos centroides $\mathbf{c}_i$, $i = 1, ..., k$ em direção a $\bar{\mathbf{x}}$, obtendo assim k novos centroides $\mathbf{c}_i^{JS}$. Hitchcock calcula cada um dos centroides $\mathbf{c}_i^{JS}$ através da seguinte equação:

$$\mathbf{c}_i^{JS} = \bar{\mathbf{x}} + \left[1 - \frac{\hat{p} - 2}{(\mathbf{c}_i - \bar{\mathbf{x}})^T \mathbf{Q}_i^{-1} (\mathbf{c}_i - \bar{\mathbf{x}})}\right]^+ (\mathbf{c}_i - \bar{\mathbf{x}}) \tag{3.1}$$

onde $\mathbf{Q}_i$ é a matriz de covariância do grupo $\mathbf{S}_i$ e $\hat{p}$ é a dimensão efetiva de $\mathbf{Q}_i$. Note que caso conheçamos as matrizes de covariância das distribuições, podemos usar estes valores reais de $\mathbf{Q}_i$ na fórmula. Caso as matrizes $\mathbf{Q}_i$ não sejam conhecidas, podemos utilizaremos as matrizes de covariância $\hat{\mathbf{Q}}_i$ dos dados pertencentes aos grupos $\mathbf{S}_i$ no lugar de $\mathbf{Q}_i$ na equação.

Após a obtenção dos pontos $\mathbf{c}_i^{JS}$, os utilizaremos como novos centroides para a execução de uma nova iteração do k-médias. A linha de execução desse algoritmo, então, é executar uma iteração do k-médias (uma iteração inclui os passos de classificação dos dados e atualização dos centroides) para obter os centroides iniciais $\mathbf{c}_i$. Após esse passo, deslocamos esses centroides em direção à média global $\bar{\mathbf{x}}$, obtendo os novos centroides $\mathbf{c}_i^{JS}$, e executamos uma nova iteração do k-médias utilizando os novos centroides $\mathbf{X}_i^{JS}$.

Porém, o autor destaca um problema que pode acontecer no método. Quando deslocamos os centroides em direção à média global, há a possibilidade de um par de

centroides coincidir, caso onde o k-médias retornaria um erro. Para contornar esse problema, gera-se um pequeno ruído (ou *jitter*) para criar uma pequena diferença entre os centroides.

Por fim, repetimos os dois últimos passos acima até que a convergência seja alcançada. O critério de convergência escolhido pelo autor foi quando o *Rand Index* entre duas configurações consecutivas do método for igual a 1, além de limitar em 10 o número máximo de iterações. Com isso, podemos resumir o funcionamento do *Hitchcock JS k-means* a partir do Algoritmo 3.

Algoritmo 3: *Hitchcock JS k-means*

1 escolher os centroides iniciais $\mathbf{c}_1, ..., \mathbf{c}_k$
2 executar uma iteração do algoritmo k-médias, obtendo novos centroides $\mathbf{c}_1, ..., \mathbf{c}_k$
3 comprimir cada um dos centroides $\mathbf{c}_i$ em direção à média global $\bar{\mathbf{x}}$, como descrito na equação (3.1), obtendo novos centroides $\mathbf{c}_1^{JS}, ..., \mathbf{c}_k^{JS}$; caso haja uma coincidência entre os centroides, aplicar um ruído sobre os centroides para diferenciá-los
4 executar uma nova iteração do algoritmo k-médias utilizando os centroides $\mathbf{c}_1^{JS}, ..., \mathbf{c}_k^{JS}$, obtendo novos centroides $\mathbf{c}_1, ..., \mathbf{c}_k$
5 repetir os passos 3 e 4 até que haja convergência

3.3 Método proposto: *Shrinkage k-means*

O método proposto, denominado *Shrinkage k-means*, tem como base a estimação dos centroides através do estimador de James-Stein. Esta ideia foi inicialmente apresentada no trabalho de Hitchcock (GAO; HITCHCOCK, 2010) e resultou no algoritmo *Hitchcock JS k-means*, descrito na Seção 3.2. O método proposto difere do *Hitchcock JS k-means* em dois pontos: estimação da matriz de covariância $\mathbf{Q}$ e escolha da aproximação $\bar{\boldsymbol{\theta}}$, que definirá a direção do encolhimento da média *MLE*.

3.3.1 Estimação da matriz de covariância

Conforme visto na Seção 2.2.1, o estimador de James-Stein foi proposto inicialmente visando a estimação da média de uma distribuição a partir de uma única amostra. Este problema, entretanto, é diferente do problema de estimação de uma média a partir de um conjunto de dados, sendo este o problema a ser resovido pelo *Hitchcock JS k-means*.

Uma adaptação do problema de estimação a partir de um conjunto de dados para um problema de estimação a partir de um único dado pode ser feita considerando a média estimada pelo *MLE*, encolhendo esta média em direção a uma aproximação. Esta média, por sua vez, não é uma amostra da distribuição, portanto não podemos utilizar a matriz de covariância da distribuição como fazemos no estimador JS. Devemos, então, utilizar

uma versão do estimador de James-Stein que permita o uso da média das amostras de uma distribuição.

Para isso, utilizaremos a ideia apontada por Efron e Morris (EFRON; MORRIS, 1973). Consideremos o problema inicial do estimador JS, que faz uma estimativa $\hat{\boldsymbol{\theta}}$ da média $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)$ da distribuição baseado numa única amostra $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2)$, onde $\mathbf{x} = (x_1, ..., x_p)$ e $x_i \sim \mathcal{N}(\theta_i, \sigma^2)$.

Porém, também temos de considerar que cada x_i pode ser uma média de n outras amostras y_{ij} de uma distribuição, com $y_{ij} \sim \mathcal{N}(\theta_i, \sigma^2)$ e $j = 1, ..., n$. Neste caso, teremos $x_i \sim \mathcal{N}(\theta_i, \sigma^2/n)$ e, consequentemente, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2/n)$. Note que, deste modo, $\mathbf{x}$ pode ser considerado um vetor da média *MLE* de n amostras de uma distribuição.

Adaptando este caso, consideremos agora o problema de estimar a média $\boldsymbol{\theta}$ a partir de uma única amostra $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, (\sigma^2/n) \cdot \mathbf{I})$. Podemos apresentar uma versão inicial do estimador de James-Stein para este caso da seguinte forma:

$$\hat{\boldsymbol{\theta}}_{JS} = \left(1 - \frac{(\sigma^2/n)(p-2)}{||\mathbf{x}||^2}\right) \cdot \mathbf{x} \tag{3.2}$$

A partir da equação acima, obtemos uma versão final do estimador de James-Stein para nosso problema. Para isso, basta a nós aplicar as derivações apresentadas na seção 2.2.4. Por fim, obtemos a seguinte equação:

$$\hat{\boldsymbol{\theta}}_{JS} = \bar{\boldsymbol{\theta}} + \left(1 - \frac{(\hat{p}-2)/n}{\left(\mathbf{x} - \bar{\boldsymbol{\theta}}\right)^T \mathbf{Q}^{-1} \left(\mathbf{x} - \bar{\boldsymbol{\theta}}\right)}\right)^+ \left(\mathbf{x} - \bar{\boldsymbol{\theta}}\right) \tag{3.3}$$

onde $\mathbf{x}$ é a média *MLE* do conjunto de dados, $\bar{\boldsymbol{\theta}}$ é a aproximação da média utilizada, $\mathbf{Q}$ é a matriz de covariância do conjunto de dados e $\hat{p}$ é a dimensão efetiva de $\mathbf{Q}$. Essa alteração na fórmula nos apresenta uma versão mais geral do estimador de James-Stein, versão essa que podemos utilizar tanto em casos com várias amostras como no caso inicial do estimador, com apenas uma amostra. Note que se aplicarmos as condições iniciais de $\bar{\boldsymbol{\theta}} = \mathbf{0}$, $\mathbf{Q} = \sigma^2\mathbf{I}$ e $n = 1$, retornamos à versão inicial do estimador de James-Stein.

É interessante verificar que para grandes valores de n, a estimação a partir desta equação resulta em um *shrinkage* pequeno, mantendo a estimativa obtida pelo JS próxima à estimativa obtida pelo *MLE* e aproveitando a precisão desta estimativa.

3.3.2 Escolha do ponto de *shrinkage*

Outro problema notado na abordagem utilizada pelo *Hitchcock JS k-means* é o uso de uma mesma aproximação da média para todos os grupos. Por exemplo, em um

caso com dois grupos bidimensionais de médias $(-10, -10)$ e $(10, 10)$, não parece correto afirmar que $(0, 0)$ é uma boa estimativa da média para os grupos.

O uso desta aproximação pelo *Hitchcock JS k-means* resulta em um problema quando tratamos de grupos desbalanceados. Considerando uma situação com dois grupos muito desbalanceados, podemos ter a média global dentro do grupo com mais elementos. Desta forma, ao fazermos o *shrinkage* o centroide do grupo com menos elementos pode ser deslocado em direção ao grupo com mais elementos, resultando em erros no agrupamento.

Com base nisto, utilizamos aproximaçãoes de média diferentes para grupos diferentes. Para cada grupo, optamos por utilizar sua mediana como aproximação para o *shrinkage* do seu centroide. No caso multidimensional, a mediana de um grupo será calculada a partir das medianas de cada uma das p dimensões dos dados.

A escolha da mediana como aproximação da média se justifica pelas suas características. Podemos afirmar que a mediana é uma boa estimativa da média, e cuja escolha como aproximação da média para o estimador JS é crucial para o método. A partir do estimador JS, quão melhor for a aproximação escolhida, melhores serão os resultados obtidos, então a escolha da mediana deve resultar em melhores resultados se comparada à média global. Outro ponto que deve ser citado sobre a mediana é sua maior tolerância a *outliers*, o que deve resultar em uma estimativa de média mais robusta pelo estimador de James-Stein.

A partir dos pontos acima, propomos uma nova abordagem para a aplicação do estimador JS ao algoritmo k-médias. A partir das alterações citadas, a abordagem proposta visa calcular cada centroide a partir do estimador JS da seguinte forma:

$$\mathbf{c}_i^{JS} = \bar{\mathbf{m}}_i + \left[1 - \frac{(\hat{p} - 2)/n_i}{(\mathbf{c}_i - \bar{\mathbf{m}}_i)^T \mathbf{Q}_i^{-1} (\mathbf{c}_i - \bar{\mathbf{m}}_i)}\right]^+ (\mathbf{c}_i - \bar{\mathbf{m}}_i) \tag{3.4}$$

onde n_i é a quantidade de elementos do grupo $\mathbf{S}_i$ e $\mathbf{m}_i$ é a mediana do grupo $\mathbf{S}_i$ (a aproximação escolhida).

A partir dessas alterações, propomos um novo método chamado de *Shrinkage k-means*. Como apresentado, o funcionamento método se assemelha ao *Hitchcock JS k-means*, diferindo na estimação da matriz de covariância $\mathbf{Q}$ e escolha da aproximação $\bar{\mathbf{\Theta}}$ a ser utilizada para o encolhimento dos centroides. Podemos resumir o funcionamento do *Shrinkage k-means* a partir do algoritmo 4.

Note que a seleção dos k centroides iniciais do algoritmo, no passo 1 do Algoritmo 4, se dá a partir de uma seleção aleatória de k pontos do conjunto de dados. Esta seleção evita que um centroide seja posicionado muito distante dos dados, e garante que cada grupo terá pelo menos um ponto, evitando, assim, a ocorrência de grupos vazios. Esta seleção também evita que tenhamos centroides iguais, visto que fazemos uma seleção de

Algoritmo 4: *Shrinkage k-means*

1 escolher os centroides iniciais $\mathbf{c}_1, ..., \mathbf{c}_k$ como k pontos aleatórios distintos do conjunto de dados
2 executar uma iteração do algoritmo k-médias, obtendo novos centroides $\mathbf{c}_1, ..., \mathbf{c}_k$
3 comprimir cada um dos centroides $\mathbf{c}_i$ em direção à mediana $\mathbf{m}_i$, como descrito na Equação (3.4), obtendo novos centroides $\mathbf{c}_1^{JS}, ..., \mathbf{c}_k^{JS}$
4 executar uma nova iteração do algoritmo k-médias utilizando os centroides $\mathbf{c}_1^{JS}, ..., \mathbf{c}_k^{JS}$, obtendo novos centroides $\mathbf{c}_1, ..., \mathbf{c}_k$
5 executar os passos 3 e 4 até que haja convergência

k pontos distintos do conjunto de dados.

4 Resultados

Com o objetivo de evidenciar as principais características do metodo proposto, inicialmente utilizamos variações de um conjunto de dados artificial. Este conjunto define um problema de classificação binário onde os grupos de dados possuem distribuição normal. A variação dessas características e seu efeito sobre o agrupamento resultante são abordados na seção 4.1.

O desempenho do *Shrinkage k-means* em problemas reais pode ser avaliado com base nos testes apresentados na seção 4.2. Nesta seção, são relatados os resultados obtidos para diferentes bases de dados reais pertencentes ao repositório da Universidade da California em Irvine (*UCI Machine Learning Repository*) (LICHMAN, 2013).

Nos testes realizados sobre dados artificiais, o método proposto foi comparado com o k-médias e o k-medianas. Tais testes visam a comparação entre as estimativas de média escolhidas, apontando a influência dessa escolha sobre o resultado em cada caso. A comparação com o k-médias é necessária para analisarmos a diferença no resultado provocada pelo uso do estimador JS. A comparação com o k-medianas é útil para analisarmos casos onde o estimador JS se aproveita de características da mediana, contrastando os resultados do uso direto da mediana com os do uso do estimador JS (que utiliza a mediana como aproximação da média).

Já nos testes realizados sobre dados reais, o método proposto também foi comparado com o método proposto por Hitchcock (que chamaremos de *Hitchcock JS k-means*). Além da comparação entre as estimativas já presente nos dados artificiais, os resultados do *Hitchcock JS k-means* serão utilizados para uma comparação entre as duas abordagens.

Para cada execução de cada um dos métodos, o *Rand Index* foi utilizado como medida de desempenho, comparando o agrupamento obtido ao agrupamento real dos dados. Em todos os gráficos são apresentados os resultados do *Rand Index* médio de 5000 execuções de cada método, trabalhando com um novo conjunto de dados gerado a cada iteração.

4.1 Dados artificiais

Os conjuntos de dados nesta seção foram gerados de modo a apontar casos onde o resultado do k-médias pode ser comprometido. Podemos afirmar que a precisão do *MLE* é comprometida em casos com poucos elementos, com poucas dimensões, com elementos dispersos ou com presença de *outliers*. Portanto, ao utilizar a média *MLE* no cálculo dos centroides, o k-médias acaba por ter alguns desses problemas.

Os testes desta seção mostram a influência de cada um desses fatores no agrupamento final, além de comparar os três métodos a fim de ressaltar a importância da escolha da estimativa da média para o agrupamento final.

Na seção 4.1.1, variamos, um por vez, a quantidade n de elementos, a quantidade p de dimensões e a variância σ dos dados, e analisamos o impacto dessas variações no resultado. Na seção 4.1.2 fazemos as mesmas variações de parâmetros, mas manipulamos a geração dos dados a fim de simular a adição de *outliers* aos grupos.

4.1.1 Variação das características do conjunto de dados

Nessa seção, geramos casos de teste semelhantes ao utilizado por Hitchcock (GAO; HITCHCOCK, 2010). O conjunto de dados gerado visa um agrupamento binário, onde cada classe é gerada a partir de uma distribuição normal p-dimensional. Essas duas distribuições terão médias $\mathbf{0}_p$ e $\mathbf{2}_p$ (onde $\mathbf{x}_p$ é o vetor p-dimensional preenchido com x) e utilizam uma mesma matriz de covariância $\mathbf{Q} = \mathbf{I}_p \cdot \sigma$. Para compor o conjunto de dados, serão geradas n amostras a partir de cada distribuição.

Para os testes a seguir, os valores de $n = 25$, $p = 50$ e $\sigma = 4$ definem o conjunto de dados padrão. Com base nesse conjunto, serão propostas variações a partir da modificação desses parâmetros, alterando-os um por vez.

O primeiro caso mostra a influência da quantidade de dados dos grupos sobre o agrupamento final. Para tanto, fixamos $p = 50$ e $\sigma = 4$ e variamos a quantidade n de elementos gerada por classe, fazendo $n = (5, 10, 15, 20, 25, 30, 35, 40, 45, 50)$. Para cada valor de n, calculamos o *Rand Index* médio de cada método.

A precisão do *MLE*, estimativa utilizada pelo k-médias, depende diretamente da quantidade de elementos, de modo que quanto mais elementos tivermos, melhor é a sua estimativa da média. A mediana também tem essa dependência, mas esta estimativa é levemente mais precisa em casos com poucos dados.

Assim, a presença de poucos elementos influencia negativamente sobre o k-médias, enquanto que o k-medianas é menos influenciado pela baixa quantidade de dados. O método proposto, por sua vez, tem um melhor desempenho para conjuntos de dados com poucos exemplos se comparados aos dois outros métodos. Esse fato é ilustrado pela Figura 4 e pelos dados referentes à figura, listados na Tabela 3.

Uma possivel explicação para o desempenho do *Shrinkage k-means* reside na utlização do estimador de James Stein para a média dos dados. Conforme apresentado anteriormente, este estimador apresenta, em média, um melhor desempenho quando comparado ao de máxima verossimilhança. Essa diferença é mais acentuada em casos como esse, onde a estimativa *MLE* não é tão precisa.

Note que à medida que aumentamos a quantidade de elementos por classe, aumentamos a precisão de todas as estimativas, fazendo com que todas se aproximem da média real e, portanto, melhorando o resultado de todos os métodos. Nesse caso, não é possível verificar grande diferença de desempenho.

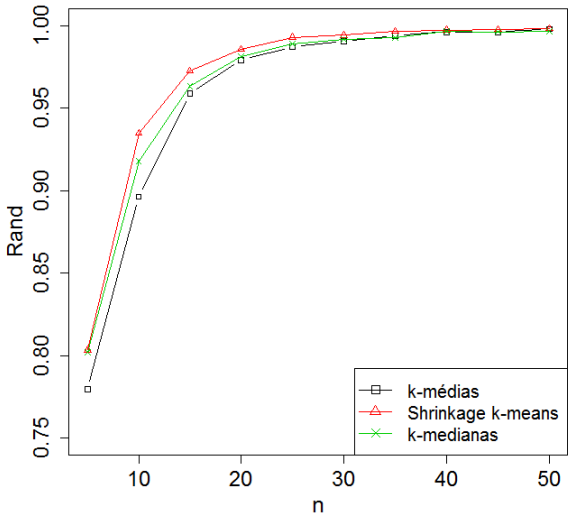

Figura 4 – Análise da variação da quantidade n de elementos (sem adição de outliers)

	k-médias	*Shrinkage k-means*	k-medianas
$n = 5$	0.7794711	0.8034711	0.8020356
$n = 10$	0.8962726	0.9347042	0.9179537
$n = 15$	0.9586998	0.9723839	0.9632662
$n = 20$	0.9791233	0.9854915	0.9810485
$n = 25$	0.9870041	0.9926581	0.9889409
$n = 30$	0.9905888	0.9940884	0.9914632
$n = 35$	0.9938968	0.9963313	0.9927049
$n = 40$	0.9960394	0.9972511	0.9966208
$n = 45$	0.9959150	0.9975329	0.9961617
$n = 50$	0.9978336	0.9981089	0.9966046

Tabela 3 – Dados referentes à análise da variação da quantidade n de elementos (sem adição de outliers), ilustrada na Figura 4

O segundo caso mostra a influência da quantidade de atributos dos dados sobre o agrupamento final. Para tanto, fixamos $n = 25$ e $\sigma = 4$ e variamos a quantidade p de dimensões dos dados, fazendo $p = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$. Para cada valor de p, calculamos o *Rand Index* médio de cada método.

Assim como no caso de poucos elementos, a precisão do *MLE* é comprometida em casos com poucas dimensões, e os três estimadores têm uma melhora nos resultados à medida que aumentamos a quantidade de dimensões. Esse fato é ilustrado pela Figura 5 e pelos dados referentes à figura, listados na Tabela 4.

A presença de poucos atributos influencia negativamente os três métodos, com um maior impacto sobre o k-medianas devido à sua estimativa, que busca minimizar o erro de cada dimensão ao selecionar um valor nela. À medida que aumentamos o número de atributos, porém, vemos uma melhora nos resultados obtidos pelos três métodos. Em todo caso, o método proposto se mostra superior aos outros dois métodos.

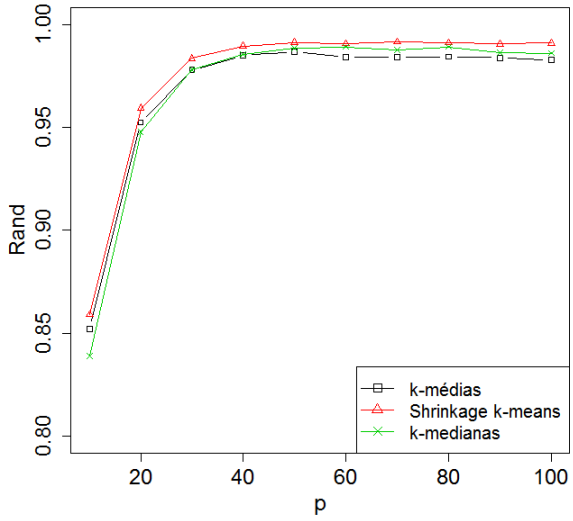

Figura 5 – Análise da variação da quantidade p de atributos (sem adição de outliers)

	k-médias	*Shrinkage k-means*	k-medianas
$p = 10$	0.8520864	0.8590318	0.8389019
$p = 20$	0.9523956	0.95919,59	0.9478735
$p = 30$	0.9779071	0.9834529	0.9779290
$p = 40$	0.9848735	0.9895239	0.9852981
$p = 50$	0.9865933	0.9911455	0.9883731
$p = 60$	0.9840829	0.9905491	0.9890784
$p = 70$	0.9839941	0.9914013	0.9874493
$p = 80$	0.9842555	0.9910010	0.9890493
$p = 90$	0.9837685	0.9904113	0.9864056
$p = 100$	0.9826880	0.9909344	0.9859816

Tabela 4 – Dados referentes à análise da variação da quantidade p de atributos (sem adição de outliers), ilustrada na Figura 5

O último caso de teste mostra a influência da dispersão dos dados no agrupamento final. Para tanto, fixamos $n = 25$ e $p = 50$ e alteramos a variância σ utilizada na geração dos dados, fazendo $\sigma = (0.1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20)$. Para cada valor de σ, calculamos o *Rand Index* médio de cada método.

A precisão do *MLE* depende diretamente da dispersão dos dados, de modo que esta média para um conjunto de dados dispersos é menos precisa do que para um conjunto

de dados aglomerados. Uma possível explicação para isso é o maior erro médio da média *MLE* em relação aos dados quando aumentamos a dispersão, o que indica uma imprecisão na estimativa da média nesse caso.

Assim como o *MLE*, a mediana também é fortemente afetada pela dispersão entre os dados. Isso acontece porque a estimativa da média pela mediana deve selecionar a mediana de cada atributo individualmente. Portanto, no caso de elementos muito dispersos, há uma maior chance de termos atributos cujas medianas não sejam precisas e cuja imprecisão aumenta à medida que aumentamos a dispersão dos dados, o que indica uma estimativa ruim.

Enquanto o aumento na variância tem um forte impacto em ambos k-médias e k-medianas, o impacto é reduzido no método proposto. Podemos atribuir esse impacto reduzido ao uso do estimador JS, que desloca a média *MLE* em direção à mediana. Apesar de estas duas estimativas demonstrarem resultados ruins em casos de alta dispersão, o uso do estimador JS no método proposto obtém um melhor resultado ao considerar a matriz de covariância dos dados no cálculo da sua estimativa. Estes fatos são ilustrados pela Figura 6 e pelos dados referentes à figura, listados na Tabela 5.

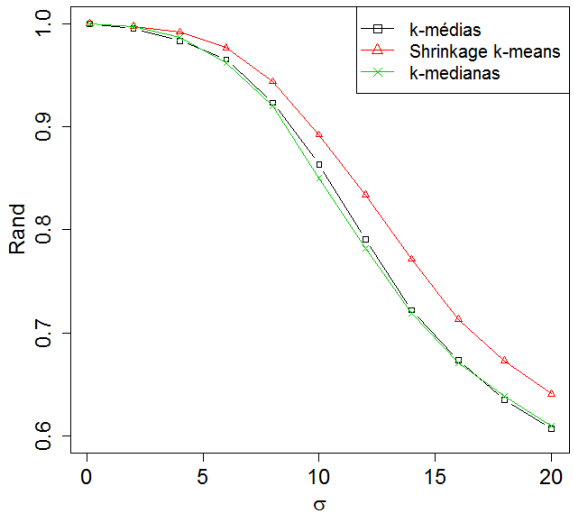

Figura 6 – Análise da alteração da variância σ (sem adição de outliers)

4.1.2 Inserção de *outliers* no conjunto de dados

Nessa seção utilizaremos um exemplo de teste semelhante ao utilizado na seção anterior, onde o conjunto de dados gerado visa um agrupamento binário, Porém, adaptamos os dados gerados de modo a adicionar *outliers* às duas classes.

Cada uma das classes p-dimensionais terá médias $\mathbf{0}_p$ e $\mathbf{2}_p$, e assim como na seção anterior, serão geradas n amostras para cada grupo. Porém, para simular a adição de

	k-médias	*Shrinkage k-means*	k-medianas
$\sigma = 0.1$	0.9996939	1.0000000	0.9998980
$\sigma = 2$	0.9952007	0.9971468	0.9969433
$\sigma = 4$	0.9830836	0.9917020	0.9866553
$\sigma = 6$	0.9651425	0.9765193	0.9621863
$\sigma = 8$	0.9230808	0.9436034	0.9198758
$\sigma = 10$	0.8631099	0.8921024	0.8499544
$\sigma = 12$	0.7906854	0.8339450	0.7819430
$\sigma = 14$	0.7221055	0.7713806	0.7188637
$\sigma = 16$	0.6737829	0.7132304	0.6710369
$\sigma = 18$	0.6353481	0.6729496	0.6384212
$\sigma = 20$	0.6075038	0.6409574	0.6096529

Tabela 5 – Dados referentes à análise da alteração da variância σ (sem adição de outliers), ilustrada na Figura 6

outliers, utilizaremos duas distribuições normais para cada classe. Para uma mesma classe, faremos com que as médias das suas distribuições sejam iguais e que as matrizes de covariância de uma das suas distribuições seja maior que a utilizada na outra distribuição.

Para gerar os dados de uma classe, utilizamos duas distribuições normais $\mathcal{N}(\mu, \mathbf{Q})$ e $\mathcal{N}(\mu, 5 \cdot \mathbf{Q})$, onde μ é a média da classe e $\mathbf{Q} = \mathbf{I}_p \cdot \sigma$. A partir destas duas distribuições, geramos as n amostras da classe, sendo que 80% serão amostras da primeira distribuição, com menor covariância, e 20% serão amostras da segunda distribuição, com maior covariância, simulando, desse modo, a adição de 20% de *outliers* para cada classe.

Para os testes a seguir, também utilizamos os valores de $n = 25$, $p = 50$ e $\sigma = 4$ para definir o conjunto de dados padrão. Assim como na seção anterior, com base nesse conjunto, serão propostas variações a partir da modificação desses parâmetros, alterando-os um por vez.

Os problemas abordados nessa seção, portanto, serão similares aos abordados na seção anterior, mas com a diferença de uma adição de 20% de *outliers* a cada classe. Podemos, assim, analisar o efeito da presença de *outliers* fazendo uma comparação com os casos da seção anterior.

No primeiro caso, fixamos $p = 50$ e $\sigma = 4$ e variamos a quantidade n de elementos gerada por classe, fazendo $n = (5, 10, 15, 20, 25, 30, 35, 40, 45, 50)$. Para cada valor de n, calculamos o *Rand Index* médio de cada método. Este caso é ilustrado pela Figura 7 e pelos dados referentes à figura, listados na Tabela 6.

No segundo caso, fixamos $n = 25$ e $\sigma = 4$ e variamos a quantidade p de dimensões dos dados, fazendo $p = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$. Para cada valor de p, calculamos o *Rand Index* médio de cada método. Este caso é ilustrado pela Figura 8 e pelos dados referentes à figura, listados na Tabela 7.

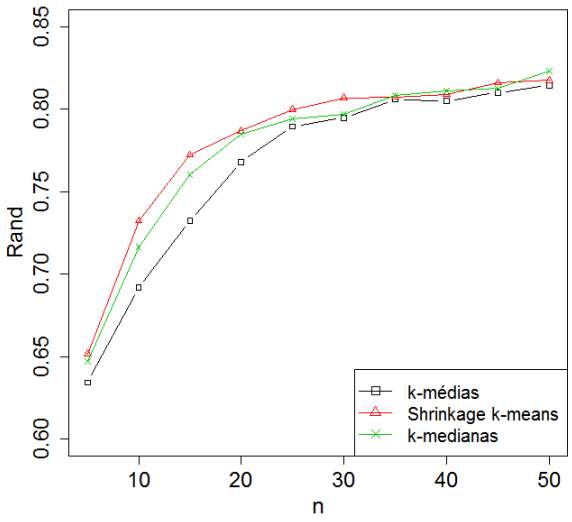

Figura 7 – Análise da variação da quantidade n de elementos (com adição de outliers)

	k-médias	*Shrinkage k-means*	k-medianas
$n = 5$	0.6342444	0.6515422	0.6472711
$n = 10$	0.6917326	0.7322495	0.7166432
$n = 15$	0.7324147	0.7720998	0.7602014
$n = 20$	0.7678754	0.7867454	0.7848923
$n = 25$	0.7893425	0.7995949	0.7939505
$n = 30$	0.7946707	0.8066136	0.7969173
$n = 35$	0.8058078	0.8070239	0.8083522
$n = 40$	0.8048331	0.8088997	0.8108848
$n = 45$	0.8099170	0.8157983	0.8123661
$n = 50$	0.8144118	0.8172598	0.8230480

Tabela 6 – Dados referentes à análise da variação da quantidade n de elementos (com adição de outliers), ilustrada na Figura 7

No último caso de teste, fixamos $n = 25$ e $p = 50$ e alteramos a variância σ utilizada na geração dos dados, fazendo $\sigma = (0.1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20)$. Para cada valor de σ, calculamos o *Rand Index* médio de cada método. Este caso é ilustrado pela Figura 9 e pelos dados referentes à figura, listados na Tabela 8.

Baseado nas figuras 7, 8 e 9, podemos concluir que o método proposto é menos sensível a *outliers* se comparado ao k-médias. Uma possível explicação para esse fato é apontada pelo uso do estimador JS e pela escolha da aproximação utilizada. Note que em todos os casos com inserção de *outliers*, o k-medianas retornou melhores resultados se comparado ao k-médias, fato justificado pela robustez da mediana (e à sensibilidade do *MLE*) a *outliers*. Assim, ao utilizar a mediana como aproximação da média o estimador JS obtém melhores resultados.

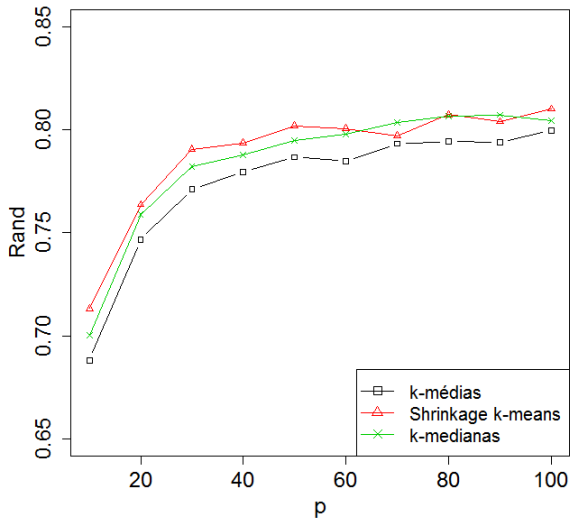

Figura 8 – Análise da variação da quantidade p de atributos (com adição de outliers)

	k-médias	*Shrinkage k-means*	k-medianas
$p = 10$	0.6881249	0.7132875	0.7005190
$p = 20$	0.7467056	0.7637146	0.7589283
$p = 30$	0.7712296	0.7903897	0.7822056
$p = 40$	0.7793887	0.7935866	0.7876090
$p = 50$	0.7866890	0.8018167	0.7947732
$p = 60$	0.7846898	0.8005251	0.7979148
$p = 70$	0.7932090	0.7970323	0.8034518
$p = 80$	0.7942547	0.8072743	0.8064054
$p = 90$	0.7937838	0.8039104	0.8071430
$p = 100$	0.7996789	0.8099530	0.8042811

Tabela 7 – Dados referentes à análise da variação da quantidade p de atributos (com adição de outliers), ilustrada na Figura 8

4.2 Dados reais

Nesta seção, analisaremos o desempenho do *Shrinkage k-means* em problemas reais, comparando o método proposto a k-médias, k-medianas e *Hitchcock JS k-means*. Os resultados relatados foram obtidos por testes usando diferentes bases de dados reais pertencentes ao repositório da Universidade da California em Irvine (*UCI Machine Learning Repository*) (LICHMAN, 2013).

Para cada base de dados utilizada, fazemos uma seleção aleatória de uma porcentagem dos dados de cada classe, formando um novo conjunto de dados (que será um subconjunto da base de dados inicial) e agrupando-o utilizando os quatro métodos. Para a seleção do subconjunto a ser utilizado, selecionamos valores de porcentagem em $(20\%, 40\%, 60\%, 80\%, 100\%)$.

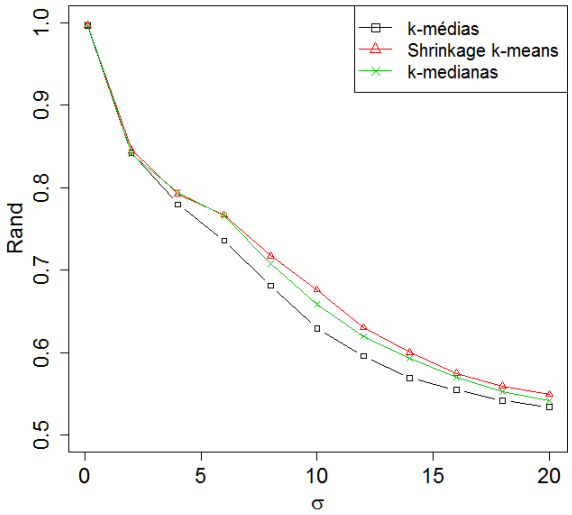

Figura 9 – Análise da alteração da variância σ (com adição de outliers)

	k-médias	*Shrinkage k-means*	k-medianas
$\sigma = 0.1$	0.9960204	0.9967347	0.9962245
$\sigma = 2$	0.8366465	0.8490815	0.8449109
$\sigma = 4$	0.7863623	0.8019659	0.7949154
$\sigma = 6$	0.7389158	0.7711918	0.7627990
$\sigma = 8$	0.6805561	0.7199507	0.7119367
$\sigma = 10$	0.6332052	0.6728873	0.6569220
$\sigma = 12$	0.5951270	0.6339285	0.6201089
$\sigma = 14$	0.5705353	0.5989494	0.5893066
$\sigma = 16$	0.5550616	0.5781597	0.5698201
$\sigma = 18$	0.5430149	0.5586261	0.5531306
$\sigma = 20$	0.5344651	0.5450046	0.5428222

Tabela 8 – Dados referentes à análise da alteração da variância σ (com adição de outliers), ilustrada na Figura 9

Para cada valor de porcentagem, geramos um novo subconjunto a cada iteração e calculamos o *Rand Index* médio de cada método. Como selecionamos uma porcentagem da quantidade total de dados de um grupo, utilizamos o piso do valor obtido a fim de evitar valores decimais de quantidades de dados a selecionar.

Com esse novo conjunto em mãos, o agrupamos utilizando o método proposto e os métodos k-médias, k-medianas e *Hitchcock JS k-means*. Por fim, calculamos, para cada método, o *Rand Index* médio entre o agrupamento obtido pelo método e a classificação real dos dados.

4.2.1 Iris

A base de dados Iris (FISHER, 1936) contém características de flores de três gêneros de plantas Iris. Seus dados estão dispostos em 3 classes, onde cada classe representa um gênero da planta. Cada classe possui 50 instâncias, e cada instância é formada por 4 características das flores desta planta. Os resultados dos testes sobre essa base de dados são ilustrados pela Figura 10 e pelos dados referentes à figura, listados na Tabela 9.

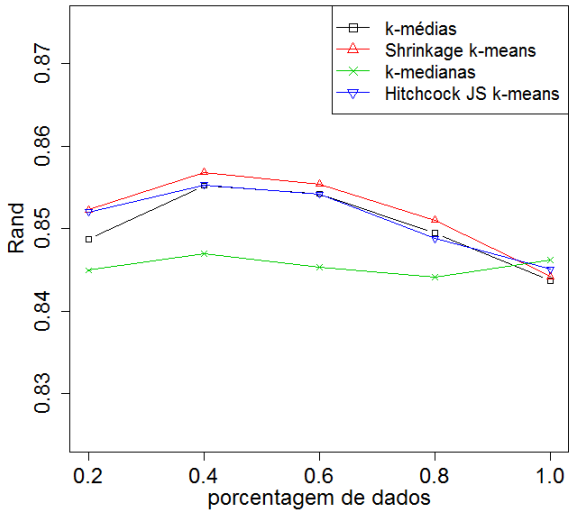

Figura 10 – Análise dos testes realizados sobre a base de dados Iris

	20%	40%	60%	80%	100%
k-médias	0.8494943	0.8546209	0.8521212	0.8489227	0.8442406
Shrinkage k-means	0.8532952	0.8563951	0.8561261	0.8507632	0.8462525
k-medianas	0.8455126	0.8448923	0.8478482	0.8458633	0.8458044
Hitchcock JS k-means	0.8520639	0.8553351	0.8542020	0.8488110	0.8451192

Tabela 9 – Dados referentes à análise dos testes realizados sobre a base de dados Iris, ilustrados na Figura 10

Essa base de dados representa um problema onde uma das classes é linearmente separável das outras duas, enquanto que essas duas não são linearmente separáveis entre si. Pela imagem, podemos concluir que o *Shrinkage k-means* mantém um ganho aproximadamente constante sobre o k-médias, além de ter um resultado médio melhor que ambos k-médias e k-medianas para qualquer valor de porcentagem.

Já se comparado ao *Hitchcock JS k-means*, o método proposto é apenas um pouco melhor. Podemos justificar esse fato pela análise do conjunto de dados. O Iris nos apresenta um problema com poucos grupos e com quantidades balanceadas de dados nas classes. Portanto, é esperado do *Hitchcock JS k-means* um resultado próximo ao do

método proposto, fato que é confirmado ao analisarmos as curvas dos dois métodos na figura 10.

4.2.2 Libras

A base de dados Libras (DIAS et al., 2009) contém características relacionadas ao movimento das mãos na Linguagem Brasileira de Sinais (LIBRAS). Para a obtenção dos dados, há um trabalho de pré-processamento de vídeo seguido de uma operação de mapeamento das informações. Os dados estão dispostos em 15 classes, onde cada classe se refere a um tipo de movimento de mãos na LIBRAS. Cada classe possui 24 instâncias, e cada instância é formada por 90 atributos. Os resultados dos testes sobre essa base de dados são ilustrados pela Figura 11 e pelos dados referentes à figura, listados na Tabela 10.

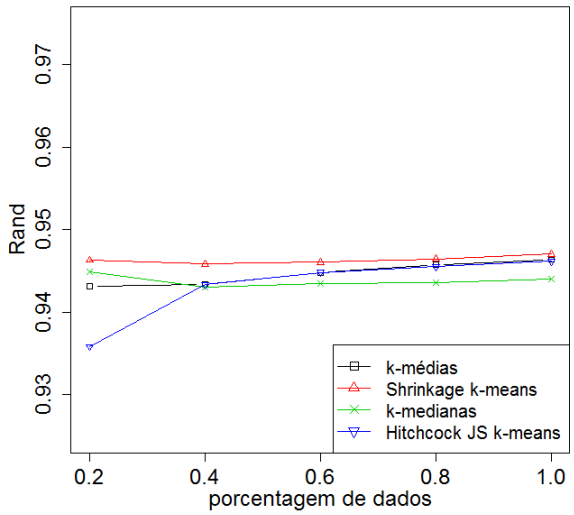

Figura 11 – Análise dos testes realizados sobre a base de dados Libras

	20%	40%	60%	80%	100%
k-médias	0.943161	0.943365	0.944787	0.945746	0.946387
Shrinkage k-means	0.9463056	0.9458512	0.9460780	0.9464596	0.9470431
k-medianas	0.9449162	0.9430798	0.9435231	0.9436129	0.9440484
Hitchcock JS k-means	0.9358421	0.9433487	0.9448481	0.9455623	0.9461909

Tabela 10 – Dados referentes à análise dos testes realizados sobre a base de dados Libras, ilustrados na Figura 11

Essa base de dados representa um problema de separação em vários grupos e com uma alta dimensionalidade dos dados. Pela imagem, podemos concluir que o *Shrinkage k-means* tem um resultado médio melhor que os outros dois métodos. Podemos notar também que os métodos, individualmente, têm médias de *Rand Index* próximas para todos

os valores de porcentagem. Podemos explicar esse fato ao analisarmos a grande quantidade de dimensões dessa base de dados, o que, como citamos anteriormente, melhora a precisão dos estimadores.

Quando comparado ao *Hitchcock JS k-means*, o método proposto obtém melhores resultados para todos os valores de porcentagens. Novamente, isso pode ser explicado pelo conjunto de dados utilizado. No Libras, temos uma grande quantidade de classes, o que provoca um resultado ruim para o *Hitchcock JS k-means*. Pela figura 11, podemos afirmar que os resultados deste método não representam um ganho considerável de desempenho se comparado ao k-médias.

4.2.3 Seeds

A base de dados Seeds (KULCZYCKI; CHARYTANOWICZ, 2011) contém características de sementes de três variedades de trigo. Seus dados estão dispostos em 3 classes, onde cada classe representa uma variedade. Cada classe possui 70 instâncias, e cada instância é formada por 7 características obtidas a partir de medidas da semente. Os resultados dos testes sobre essa base de dados são ilustrados pela Figura 12 e pelos dados referentes à figura, listados na Tabela 11.

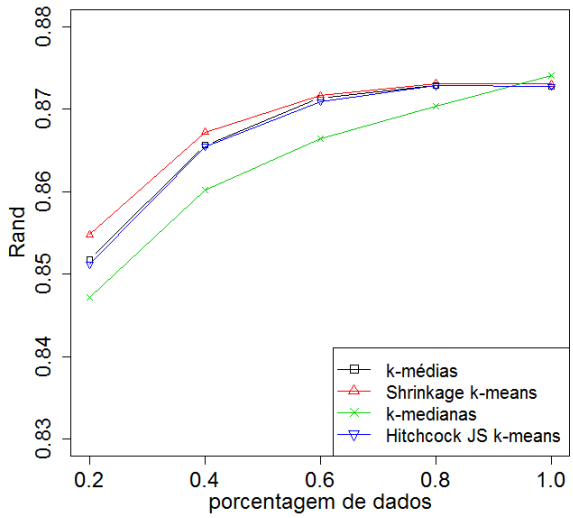

Figura 12 – Análise dos testes realizados sobre a base de dados Seeds

Essa base de dados representa um problema de separação com poucas dimensões e com grande quantidade de dados por grupo. Pela imagem, podemos concluir que o método proposto tem um resultado médio melhor que os outros dois métodos. No caso do *Hitchcock JS k-means*, vemos que não há uma vantagem considerável em relação ao k-médias. Também pela imagem, podemos detectar o efeito quantidade de elementos sobre o resultado obtido.

	20%	40%	60%	80%	100%
k-médias	0.8516815	0.8653726	0.8707416	0.8726445	0.8726455
Shrinkage k-means	0.8547789	0.8671920	0.8710162	0.8731538	0.8729085
k-medianas	0.8465816	0.8618772	0.8668423	0.8704488	0.8736823
Hitchcock JS k-means	0.8512118	0.8654517	0.8708801	0.8728088	0.8727439

Tabela 11 – Dados referentes à análise dos testes realizados sobre a base de dados Seeds, ilustrados na Figura 12

Um ponto a ser analisado é a diferença entre os resultados de *Shrinkage k-means* e k-médias, que é reduzida à medida que aumentamos a quantidade de amostras. Para isso, basta-nos analisar a fórmula de atualização dos centroides do método proposto.

$$\mathbf{c}_i^{JS} = \bar{\mathbf{m}}_i + \left[1 - \frac{(\hat{p}-2)/n_i}{(\mathbf{c}_i - \bar{\mathbf{m}}_i)^T \mathbf{Q}_i^{-1} (\mathbf{c}_i - \bar{\mathbf{m}}_i)}\right]^+ (\mathbf{c}_i - \bar{\mathbf{m}}_i)$$

Pela fórmula acima, notamos que à medida que aumentamos a quantidade n_i de amostras de um grupo, fazemos com que nossa estimativa da média desse grupo se aproxime do centroide $\mathbf{c}_i$, obtido pela estimativa *MLE*. Esse é um caso no qual faz sentido nos aproximarmos do *MLE*, visto que esta estimativa se torna mais precisa à medida que aumentamos o número de amostras.

4.2.4 Wine

A base de dados Wine (FORINA et al., 1988) contém características de três tipos de vinhos. Seus dados estão dispostos em 3 classes, onde cada classe representa um tipo de vinho. As três classes são desbalanceadas, contendo 59, 71 e 48 amostras cada. Cada instância é formada por 13 características obtidas a partir de análises químicas dos vinhos. Os resultados dos testes sobre essa base de dados são ilustrados pela Figura 13 e pelos dados referentes à figura, listados na Tabela 12.

	20%	40%	60%	80%	100%
k-médias	0.7105961	0.7098942	0.7126901	0.7129288	0.7125510
Shrinkage k-means	0.7117854	0.7106752	0.7130027	0.7132995	0.7122833
k-medianas	0.7148606	0.7154531	0.7173748	0.7177067	0.7242858
Hitchcock JS k-means	0.7120963	0.7106546	0.7121225	0.7130320	0.7124896

Tabela 12 – Dados referentes à análise dos testes realizados sobre a base de dados Wine, ilustrados na Figura 13

Essa base de dados representa um problema com classes desbalanceadas e com alta correlação entre as variáveis. Nesse caso, o k-medianas se destaca com os melhores resultados, enquanto que o método proposto tem um resultado próximo ao obtido pelo

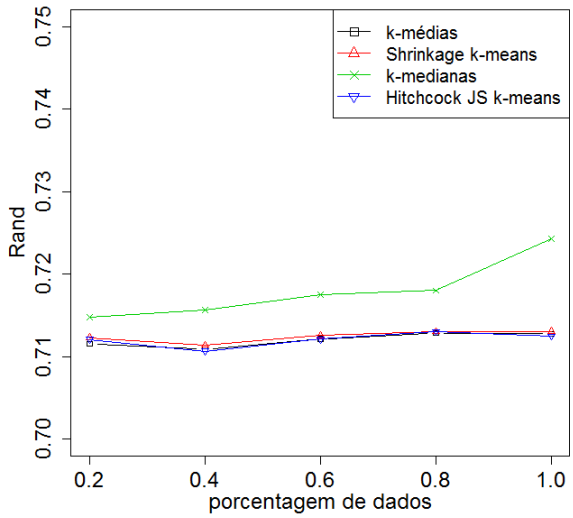

Figura 13 – Análise dos testes realizados sobre a base de dados Wine

k-médias. Esse fato também pode ser explicado ao analisarmos a fórmula de atualização dos centroides do método proposto.

No caso de dados com alta correlação, teremos um valor muito baixo de $\hat{p}$ (dimensão efetiva de $\mathbf{Q}_i$). Ao analisarmos a fórmula, podemos afirmar que quanto menor o valor de $\hat{p}$, maior é a proximidade da nossa estimativa de média do grupo em relação ao centroide $\mathbf{c}_i$, obtido pela estimativa *MLE*. Isso também explica os resultados do *Hitchcock JS k-means*, visto que este método utiliza a dimensão efetiva $\hat{p}$ de um modo semelhante ao nosso método.

5 Conclusão

5.1 Considerações finais

O processo de agrupamento do método k-médias busca reconhecer k distribuições normais multivariadas dentro do conjunto de dados, onde cada distribuição representará um grupo. Ao agrupar um elemento de acordo com as distâncias desse elemento a cada um dos centroides, o k-médias considera que todas as distribuições têm uma mesma matriz de covariância. Baseado nisso, o objetivo do método é estimar a média das k distribuições para encontrar um melhor posicionamento dos grupos.

A partir de uma análise do estimador de James-Stein e da sua vantagem em relação ao *MLE*, utilizado pelo k-médias, foi proposta uma modificação desse método. Esta modificação originou o algoritmo *Shrinkage k-means*, que busca utilizar o estimador de James-Stein e usufruir da sua vantagem em relação ao *MLE* a fim de obter melhores agrupamentos dos dados. Para essa aplicação, escolhemos a mediana como estimativa a ser utilizada como aproximação da média pelo estimador de James-Stein.

Utilizando como base as análises e os experimentos realizados neste trabalho, conclui-se que o *Shrinkage k-means* apresenta melhores resultados do que os obtidos pelo k-médias, sendo, no pior dos casos, tão bom quanto o k-médias. O uso da mediana, estimativa robusta a *outliers*, como aproximação da média faz com que o *Shrinkage k-means* obtenha resultados melhores em casos com presença de *outliers*. O uso do estimador de James-Stein também se beneficia da precisão do *MLE* em casos com muitos elementos, caso onde sua estimativa de média será muito próxima à do *MLE*.

O *Shrinkage k-means* também apresenta melhores resultados se comparado ao método *Hitchcock JS k-means*, anteriormente proposto por Hitchcock. Essa melhora é provocada pela alteração na fórmula de atualização dos centroides, que passa a considerar a quantidade de elementos no seu cálculo.

5.2 Trabalhos futuros

Com relação ao desenvolvimento de trabalhos futuros, sugere-se a aplicação do estimador de James-Stein a outros métodos de agrupamento particional, como *fuzzy c-means*. A ideia dessa aplicação é avaliar a precisão do estimador de James-Stein quando comparado a outras estimativas de média utilizadas, baseando essa avaliação na comparação dos agrupamentos resultantes dos métodos.

Pode-se também utilizar o método proposto em outros algoritmos que utilizem o

k-médias, como o *Spectral*, que inclui em seu algoritmo uma execução do k-médias. A ideia desse uso é que uma vez efetuada essa substituição do método utilizado, espera-se que resultados melhores sejam obtidos pelo algoritmo modificado.

Sugere-se também a análise dos resultados quando da substituição da mediana por outra aproximação da média a ser utilizada pelo estimador de James-Stein, como medóide ou média aparada. Tais testes devem demonstrar os ganhos obtidos quando do uso de cada estimativa, visto que o estimador de James-Stein acaba por manter algumas das vantagens das aproximações por ele utilizadas.

Referências

ARTHUR, D.; VASSILVITSKII, S. K-means++: The advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007. (SODA '07), p. 1027–1035. ISBN 978-0-898716-24-5.

BARANCHIK, A. *Multiple Regression and Estimation of the Mean of a Multivariate Normal Distribution*. [S.l.]: Department of Statistics, Stanford University, 1964. (Technical report).

BERKHIN, P. A survey of clustering data mining techniques. In: KOGAN, J.; NICHOLAS, C.; TEBOULLE, M. (Ed.). *Grouping Multidimensional Data*. [S.l.]: Springer Berlin Heidelberg, 2006. p. 25–71. ISBN 978-3-540-28348-5.

BOCK, M. E. Minimax estimators of the mean of a multivariate normal distribution. *Ann. Statist.*, The Institute of Mathematical Statistics, v. 3, n. 1, p. 209–218, 01 1975.

DIAS, D. et al. Hand movement recognition for brazilian sign language: A study using distance-based neural networks. In: *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. [S.l.: s.n.], 2009. p. 697–704. ISSN 1098-7576.

EFRON, B.; MORRIS, C. Stein's Estimation Rule and Its Competitors–An Empirical Bayes Approach. *Journal of the American Statistical Association*, American Statistical Association, v. 68, n. 341, p. 117–130, 1973. ISSN 01621459.

EFRON, B.; MORRIS, C. Data Analysis Using Stein's Estimator and its Generalizations. *Journal of the American Statistical Association*, American Statistical Association, v. 70, n. 350, p. 311–319, jun. 1975.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, v. 7, n. 7, p. 179–188, 1936.

FORINA, M. et al. Parvus - an extendible package for data exploration, classification and correlation. *Journal of Chemometrics*, John Wiley & Sons, Ltd., v. 4, n. 2, p. 191–193, 1988.

FU, K.; MUI, J. A survey on image segmentation. *Pattern Recognition*, v. 13, n. 1, p. 3 – 16, 1981. ISSN 0031-3203.

GAO, J.; HITCHCOCK, D. B. James-stein shrinkage to improve k-means cluster analysis. *Comput. Stat. Data Anal.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 54, n. 9, p. 2113–2127, set. 2010. ISSN 0167-9473.

GOWER, J. C.; ROSS, G. J. S. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Wiley for the Royal Statistical Society, v. 18, n. 1, p. pp. 54–64, 1969. ISSN 00359254.

JAIN, A.; DUIN, R.; MAO, J. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 22, n. 1, p. 4–37, Jan 2000. ISSN 0162-8828.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.

JAMES, W.; STEIN, C. Estimation with quadratic loss. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Berkeley, Calif.: University of California Press, 1961. p. 361–379.

KAUFMAN, L.; ROUSSEEUW, P. Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods*, p. North–Holland, 1987.

KAUFMAN, L.; ROUSSEEUW, P. J. Agglomerative nesting (program AGNES). In: *Finding Groups in Data: An Introduction to Cluster Analysis*. [S.l.]: John Wiley & Sons, Inc., 1990. p. 199–252. ISBN 9780470316801.

KAUFMAN, L.; ROUSSEEUW, P. J. Divisive analysis (program DIANA). In: *Finding Groups in Data: An Introduction to Cluster Analysis*. [S.l.]: John Wiley & Sons, Inc., 1990. p. 253–279. ISBN 9780470316801.

KOCH, K. *Introduction to Bayesian Statistics*. [S.l.]: Springer Berlin Heidelberg, 2007. ISBN 9783540727262.

KULCZYCKI, P.; CHARYTANOWICZ, M. A complete gradient clustering algorithm. In: *Proceedings of the Third International Conference on Artificial Intelligence and Computational Intelligence - Volume Part III*. Berlin, Heidelberg: Springer-Verlag, 2011. (AICI'11), p. 497–504. ISBN 978-3-642-23895-6.

LICHMAN, M. *UCI Machine Learning Repository*. 2013. Disponível em: <http://archive.ics.uci.edu/ml>.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: *Procedings of the Fifth Berkeley Symposium on Math, Statistics, and Probability*. [S.l.]: University of California Press, 1967. v. 1, p. 281–297.

RAND, W. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, v. 66, n. 336, p. 846–850, 1971.

RUSPINI, E. H. A new approach to clustering. *Information and Control*, v. 15, n. 1, p. 22–32, 1969.

SELIM, S. Z.; ISMAIL, M. A. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 6, n. 1, p. 81–87, jan. 1984. ISSN 0162-8828.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 22, n. 8, p. 888–905, ago. 2000. ISSN 0162-8828.

SIBSON, R. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, v. 16, n. 1, p. 30–34, 1973.

STEIN, C. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. Berkeley, Calif.: University of California Press, 1956. p. 197–206.

STEINHAUS, H. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci. Cl. III. 4*, p. 801–804, 1956.