Combinando semi-supervisão e *hubness* para aprimorar o agrupamento de dados em alta dimensão

Mateus C. de Lima[1], **Maria Camila N. Barioni**[1], **Humberto L. Razente**[1]

[1]Faculdade de Computação
Universidade Federal de Uberlândia (FACOM/UFU) – Uberlândia, MG – Brasil

mateuscurcino@mestrado.ufu.br, {camila.barioni,humberto.razente}@ufu.br

Abstract. *The curse of dimensionality turns the high-dimensional data analysis a challenging task for data clustering techniques. In order to deal with high-dimensional data, this paper presents a clustering approach that explores the combination of two strategies: semi-supervision and density estimation based on hubness scores. Initial experimental results show a good performance when applied on real data sets with different characteristics.*

Resumo. *A chamada maldição da dimensionalidade faz com que a análise de dados em alta dimensão seja uma tarefa desafiadora para técnicas de agrupamento de dados. Este artigo apresenta uma abordagem de agrupamento que explora a combinação de estratégias de semi-supervisão e de estimativa de densidade baseada em pontuações hubness com foco em dados de alta dimensão. Os resultados experimentais iniciais mostram o seu bom desempenho quando aplicada em conjuntos de dados reais com diferentes características.*

1. Introdução

Existem vários domínios de aplicação para os quais o emprego de técnicas de mineração de dados é útil em que a dimensionalidade dos dados é notavelmente elevada. Dentre eles estão bancos de dados onde cada instância de dados é descrita por uma coleção de características, como no caso de imagens e de expressões gênicas. A chamada maldição da dimensionalidade [Samet 2005] faz com que a análise de dados em alta dimensão seja uma tarefa desafiadora para qualquer técnica de mineração de dados baseada em cálculos de distância, uma vez que a medida que a dimensionalidade aumenta, também aumenta a esparsidade dos dados e a dificuldade de diferenciar instâncias de dados. Nesse contexto, é interessante considerar o emprego de técnicas que permitam atenuar esses efeitos prejudiciais a eficiência e a eficácia de algoritmos de mineração de dados.

Tradicionalmente, a questão da alta dimensionalidade tem sido tratada na literatura científica da área com a utilização de estratégias que permitem realizar a redução de dimensionalidade por meio de extração e de seleção de atributos [Faceli et al. 2011]. Entretanto, informações podem ser perdidas quando a dimensionalidade é reduzida [Silvestre 2007]. De uma maneira oposta, estratégias mais recentes têm empregado de maneira eficaz um aspecto inerente aos dados de alta dimensão na proposta de técnicas que permitem a classificação [Tomasev and Mladenic 2013] e o agrupamento [Tomasev et al. 2014] em bancos de dados de alta dimensão. Esse aspecto, denominado

Trabalho realizado com apoio financeiro do CNPq e da CAPES.

hubness, consiste na tendência de algumas instâncias de dados, chamadas *hubs*, ocorrerem com maior frequência nas listas dos K-vizinhos mais próximos de outras instâncias.

Apesar dessas técnicas terem demonstrado que agrupamentos de dados orientados pelas informações da pontuação *hubness* das instâncias de dados apresentam bons resultados quando aplicados a dados de alta dimensão, é importante destacar que como os *hubs* são centros de influência, possíveis imprecisões associadas a eles podem ser facilmente propagadas [Tomasev and Mladenic 2013]. Uma estratégia que pode ser utilizada para minimizar o risco de induzir um particionamento não adequado nos dados, uma vez que *hubs* podem não refletir bem a semântica implícita dos dados, é a incorporação de semi-supervisão [Basu et al. 2008].

Este artigo apresenta a proposta de uma abordagem de agrupamento que explora a combinação de estratégias de semi-supervisão e de utilização das pontuações *hubness* a respeito das instâncias de dados com foco em dados de alta dimensão. Essa abordagem revisitou o algoritmo *HPKM* [Tomasev et al. 2011] dando origem ao método denominado *SSHub Clustering* (*Semi-supervised hubness-based clustering*). Analisando os resultados experimentais iniciais obtidos é possível constatar que o *SSHub Clustering* obteve bom desempenho quando aplicado em conjuntos de dados reais de diferentes tamanhos e dimensões. O restante do artigo está organizado como descrito a seguir. A Seção 2 apresenta alguns conceitos fundamentais. O *SSHub Clustering* é descrito na Seção 3. A discussão a respeito dos resultados experimentais obtidos é apresentada na Seção 4. E, as conclusões e os trabalhos futuros são descritos na Seção 5.

2. Aspecto *Hubness*

Informações sobre *hubness* podem ser obtidas a partir de listas de K-vizinhos mais próximos. A medida que a dimensionalidade intrínseca dos dados aumenta a distribuição das K-ocorrências na lista de vizinhos mais próximos de cada instância de dados torna-se distorcida. Dessa forma, algumas instâncias (*hubs*) aparecem com maior frequência na listagem dos K-vizinhos mais próximos, indicando regiões mais densas, e algumas outras instâncias (*anti-hubs*), nas regiões mais esparsas, tornam-se vizinhos infrequentes [Tomasev et al. 2014].

Segundo [Tomasev et al. 2014] a pontuação *hubness* de uma dada instância de dados pode ser definida da seguinte maneira: Seja $X \subset R^d$ um conjunto de instâncias de dados, $h_K(x)$ representa o número de K-ocorrências de instâncias $x \in X$, isto é, o número de vezes que x ocorre na listagem dos K-vizinhos mais próximos de outras instâncias pertencentes a X. As instâncias de dados que possuem $h_K(x)$ significativamente acima da média são chamadas de $hubs$. Por outro lado, as que possuem $h_K(x)$ extremamente baixo correspondem aos *anti-hubs*. Existem trabalhos que mostram que a pontuação *hubness* é uma boa medida de centralidade para agrupamentos de dados de alta-dimensionalidade [Tomasev et al. 2011] [Tomasev and Mladenic 2013] [Tomasev et al. 2014].

3. *SSHub Clustering*

A partir de um conjunto de dados $X = \{x_1; ...; x_n\}$, de um conjunto H com a pontuação *hubness* $h_K(x)$ de cada instância x_i, e de um número f de elementos de fronteira a serem analisados para a definição das restrições *must-link* e *cannot-link*, o *SSHub Clustering* emprega uma abordagem de agrupamento semi-supervisionada baseada no par-

ticionamento de X em uma quantidade k de grupos. Cada grupo é representado por múltiplos protótipos, um principal $p_i \in P$ e vários auxiliares $a_i \in A$. Essa estratégia de representação foi adotada com o intuito de permitir que o algoritmo possa lidar bem tanto com grupos com formatos hiperesféricos como com grupos que apresentem formas mais complexas.

Para criar a partição inicial dos dados esse algoritmo inicia selecionando, de modo semi-supervisionado, um representante principal para cada grupo. A estratégia proposta para essa seleção consiste em guiar o usuário, a partir de um *ranking* das maiores pontuações *hubness* de cada instância de dados, na indicação de k instâncias que devam ser usadas como representantes. Os demais representantes auxiliares são derivados de restrições *must-link* $r_m(x_i, x_j) \in R_m$ a partir da segunda iteração do algoritmo.

A atribuição de instâncias aos grupos considera a menor distância agregada da instância x_i para o representante principal e os auxiliares, conforme a Equação 1:

$$\delta_g(Q_k, x_i) = \min_{q_j \in Q_k} (\delta(q_j, x_i)) \tag{1}$$

, onde Q_k é o conjunto de representantes (principal e auxiliares) de um grupo c_k e $\delta()$ é uma função de distância.

A atualização dos representantes principais de cada grupo baseia-se na pontuação *hubness* das instâncias pertecentes a cada grupo. Contudo, essa pontuação pode ser alterada ao longo das iterações do algoritmo. Inspirado na abordagem do algoritmo ***HPKM*** [Tomasev et al. 2014], instâncias que não trocam de grupo ao longo das iterações são privilegiadas elevando suas pontuações *hubness* ao quadrado. No caso de haver mudança de grupo, a pontuação *hubness* retorna ao valor original. O representante de cada grupo corresponde a instância com a maior pontuação *hubness* acumulada. Resumidamente, os passos realizados pelo ***SSHub Clustering*** são apresentados abaixo:

(1) Selecionar k instâncias com alta pontuação *hubness* como representantes iniciais $P = \{p_i, ..., p_k\}$ e fazer $A = P$;

(2) Atribuir cada instância $xi \in X$ ao grupo com menor distância agregada δ_g em relação a Q, respeitando as restrições R_m e R_c;

(3) Atualizar os representantes principais P de cada grupo com base nas pontuações *hubness* das instâncias atribuídas para cada grupo;

(4) Analisar as f instâncias de fronteira para definir restrições *must-link* e *cannot-link* e mais representantes auxiliares;

(5) Retornar ao passo (2) até atingir o número máximo de iterações permitidas.

No passo (4), considerando as instâncias de dados atribuidas a cada grupo $\pi_j \in \Pi$ deseja-se selecionar um subconjunto de instâncias de tal forma que obter conhecimento adicional sobre elas permita guiar o agrupamento dos dados para um particionamento mais adequado. Para tanto, as f instâncias mais distantes de cada $p_i \in P$ com pontuação *hubness* ≥ 1 são avaliadas. Essa restrição para a pontuação *hubness* é para evitar que possíveis *anti-hubs* sejam selecionados.

A partir da identificação de cada instância de fronteira x_y de um dado grupo π_i, seleciona-se a instância x_w de π_j mais próxima de x_y, tal que $\pi_i \neq \pi_j$. Com isso,

é possível analisar pares de instâncias de fronteira de grupos distintos e questionar se elas devem ou não estar em um mesmo grupo e gerar restrições *must-link* $r_m(x_y, x_w)$ ou restrições *cannot-link* $r_c(x_y, x_w)$, respectivamente. Além disso, para obter representantes auxiliares para os grupos, cada x_y de um dado grupo π_i é analisado para questionar se x_y e p_i devem ou não estar em um mesmo grupo. Caso a resposta seja positiva, cria-se uma $r_m(x_y, p_i)$ e atribui-se x_y a A. Caso contrário, cria-se uma $r_c(x_y, p_i)$.

4. Experimentos

Para a realização dos experimentos foram considerados 9 conjuntos de dados reais obtidos da *UCI Machine Learning*[1] e um conjunto de dados reais (NBA) disponível no repositório BasketballStats[2]. Todos esses conjuntos de dados foram pré-processados de acordo com as recomendações indicadas nesses repositórios. Os detalhes a respeito de cada um deles são apresentados na Tabela 1. É importante destacar que foram selecionados conjuntos de dados com diferentes números de classes, densidades e dimensionalidades com o intuito de verificar o desempenho do algoritmo proposto frente a outros três algoritmos descritos na literatura científica da área em diferentes cenários.

Tabela 1. Conjuntos de dados considerados nos experimentos.

Identificador	Conjunto	Instâncias	Dimensões	Classes
1	*Arrhythmia*	68	279	11
2	*Ecoli*	336	7	8
3	*Isolet*	6237	617	26
4	*LungCancer*	27	56	3
5	*NBA*	22064	17	2
6	*Pendigits*	7494	16	10
7	*Segment*	2310	19	7
8	*UrbanLandCover*	675	147	9
9	*WPBC*	569	30	2
10	*Zoo*	101	16	7

Os algoritmos *HPKM*, *Kernel k-means* [Dhillon et al. 2004] e *DBSCAN* [Sander et al. 1998] foram selecionados como linha de base para comparação. O *HPKM* foi o algoritmo de agrupamento baseado em *hubs* que inspirou a proposta do *SSHub Clustering*. O *Kernel k-means* é um representante clássico de algoritmos de agrupamento baseados em *kernel* que são conhecidos por lidarem bem com grupos não hiperesféricos e o *DBSCAN* é um representante padrão de algoritmos de agrupamento baseados em densidade.

Considerando que o tamanho ideal de vizinhança a ser analisado no cálculo da pontuação *hubness* pode variar de acordo com o domínio de dados, foram considerados quatro valores distintos de vizinhança (K = 5, 10, 15 e 20) para o *SSHub Clustering* e o *HPKM*. A quantidade de grupos solicitada (k) seguiu a quantidade de classes definida para cada conjunto de dados. Todos os algoritmos consideraram a função de distância Euclidiana.

Para automatizar a realização de experimentos com o *SSHub Clustering*, a etapa de interação com o usuário na escolha dos representantes iniciais foi simulada da seguinte maneira. Levando em consideração as informações de rótulo dos conjuntos de dados, os representantes iniciais foram sorteados entre as cinco instâncias de cada grupo com

[1] http://archive.ics.uci.edu/ml/

[2] http://www.databasebasketball.com/stats-download.htm

as maiores pontuações *hubness*. Dessa forma, é possível simular o fato de que a cada execução o usuário poderia escolher diferentes instâncias de alta pontuação *hubness* como representantes. Além disso, a quantidade total de elementos de fronteira (f) analisados pelo *SSHub Clustering* foi definida em 1% do total de instâncias, divididos igualmente entre o número de grupos solicitados.

Considerando a existência de conhecimento a respeito do particionamento desejado para os conjuntos de dados, foi utilizado o índice de validação *Rand* [Faceli et al. 2011] para avaliar a eficácia dos algoritmos. Os resultados apresentados na Tabela 2 consideram a média de 50 execuções dos algoritmos desconsiderando a melhor e a pior execução de cada um deles. Observando essa tabela é possível verificar que o algoritmo *SSHub Clustering* apresentou resultados superiores em todos os conjuntos, com exceção do conjunto (7), onde empatou com o algoritmo *DBSCAN*.

Também é possível constatar que a vizinhança definida para o cálculo da pontuação *hubness* pode impactar nos particionamentos gerados. Este fato pode ser observado, principalmente, nos conjuntos (4), (5), (9) e (10). Além disso, é possível notar uma superioridade significativa do algoritmo *SSHub Clustering*, em relação aos demais algoritmos, nos conjuntos de dados (4), (5) e (10).

Tabela 2. Resultados dos experimentos. Os valores em negrito destacam os melhores desempenhos. Os números entre () indicam a posição no *rank*.

Conjunto	*SSHub Clustering*				*HPKM*				*KERNEL*	*DBSCAN*
	$K=5$	$K=10$	$K=15$	$K=20$	$K=5$	$K=10$	$K=15$	$K=20$		
1	0,76	0,75	0,76	**0,78** (1)	0,76 (2)	0,76	0,76	0,76	0,42 (4)	0,64 (3)
2	0,87	**0,89** (1)	0,88	0,88	0,81 (2)	0,81	0,80	0,81	0,27 (4)	0,43 (3)
3	**0,96** (1)	**0,96**	**0,96**	**0,96**	0,95 (2)	0,95	0,95	0,95	0,86 (3)	0,47 (4)
4	0,55	0,60	0,69	**0,70** (1)	0,54 (2)	0,52	0,51	0,50	0,31 (4)	0,47 (3)
5	**0,77** (1)	0,65	0,66	0,70	0,52 (4)	0,52	0,52	0,52	0,60 (3)	0,65 (2)
6	**0,95** (1)	0,94	**0,95**	**0,95**	0,91 (3)	0,91	0,91	0,91	0,10 (4)	0,93 (2)
7	**0,85** (1,5)	0,83	0,84	0,84	0,80	0,80	0,80	0,81 (3)	0,16 (4)	**0,85** (1,5)
8	0,78	**0,79** (1)	0,77	0,77	0,70 (2)	0,69	0,70	0,70	0,14 (4)	0,41 (3)
9	0,80	**0,84** (1)	0,78	0,80	0,74	0,75 (2)	0,75	0,75	0,53 (4)	0,66 (3)
10	0,92	0,91	**1,00** (1)	0,93	0,88	0,89	0,88	0,90 (2)	0,77 (4)	0,83 (3)
Soma *ranks*	10,5				24				38	27,5
Média *ranks*	1,05				2,4				3,8	2,75

Para verificar se o *SSHub Clustering* supera estatisticamente os demais algoritmos testados, os desempenhos dos algoritmos foram submetidos ao teste estatístico de *Friedman* [Demsar 2006] e *post-hoc de Bonferroni-Dunn* [Zar 2007] com o objetivo de descobrir se com 90% de confiança pode-se concluir que o algoritmo *SSHub Clustering* supera os algoritmos *HPKM*, *Kernel k-means* e *DBSCAN*. Para aplicar o teste de *Friedman*, considera-se a soma dos *ranks* (R) de cada algoritmo para o cálculo de F_F como: $F_F = ((N-1)\chi_F^2)/(N(A-1)-\chi_F^2)$, onde $\chi_F^2 = (12/(NA(A+1))\left[\sum_{i=1}^{A} R_i^2 - (3NA(A+1))\right])$, N é o número de conjuntos de dados e A é o número de algoritmos testados. Como o valor de F_F (324) é superior ao valor crítico *FDistribution* (2,29), conclui-se que existe diferença significativa entre os resultados dos algoritmos.

Para indicar qual dos algoritmos teve desempenho superior foi utilizado o teste

post-hoc Bonferroni-Dunn fixando o *SSHub Clustering* como o algoritmo de controle. Neste teste, o desempenho de dois algoritmos é estatisticamente diferente se a diferença entre a média dos *ranks* é maior ou igual a diferença crítica (CD), que é calculada como: $CD = q_\alpha \sqrt{\frac{A(A+1)}{6N}}$, onde o valor crítico q_α (2,12) é obtido na tabela encontrada em [Demsar 2006]. Com o valor CD calculado (1,22) é possível observar que as diferenças entre as médias dos *ranks* dos algoritmos *SSHub Clustering* e *HPKM* (1,35), *SSHub Clustering* e *Kernel k-means* (2,75) e *SSHub Clustering* e *DBSCAN* (1,70) foram superiores a CD. Com isso, é possível afirmar que a eficácia do algoritmo *SSHub Clustering* é superior a dos algoritmos selecionados como linha de base para os conjuntos de dados testados.

5. Conclusões

O objetivo do *SSHub Clustering* é combinar o aspecto *hubness* com estratégias de semi-supervisão para permitir a obtenção de agrupamentos que condizam melhor com o particionamento esperado para dados de alta dimensão. Os resultados experimentais mostram que o método desenvolvido possui grande potencial para lidar com diferentes quantidades de instâncias, grupos e dimensões. Além disso, foi verificado que o *SSHub Clustering* supera estatisticamente três abordagens propostas na literatura que apresentam bons resultados no tratamento de dados de alta dimensão. Dentre os trabalhos futuros deseja-se realizar mais experimentos com outros conjuntos de dados reais e sintéticos comparando o *SSHub Clustering* também com algoritmos de agrupamento semi-supervisionado.

Referências

Basu, S., Davidson, I., and Wagstaff, K. (2008). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 1 edition.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: Spectral clustering and normalized cuts. KDD '04, pages 551–556. ACM.

Faceli, K., Lorena, A. C., Gama, J. a., and Carvalho, A. (2011). *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. LTC, 1 edition.

Samet, H. (2005). *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann.

Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Min. Knowl. Discov.*, 2(2):169–194.

Silvestre, A. L. (2007). *Análise de Dados e Estatística Descritiva*. Escolar Editora.

Tomasev, N. and Mladenic, D. (2013). Hub co-occurrence modeling for robust high-dimensional knn classification. In *ECML PKDD*, pages 643–659. Springer.

Tomasev, N., Radovanovic, M., Mladenic, D., and Ivanovic, M. (2011). The role of hubness in clustering high-dimensional data. PAKDD, pages 183–195. Springer.

Tomasev, N., Radovanovic, M., Mladenic, D., and Ivanovic, M. (2014). The role of hubness in clustering high-dimensional data. *IEEE TKDE*, 26(3):739–751.

Zar, J. H. (2007). *Biostatistical Analysis*. Prentice-Hall, Inc., 5 edition.