Implementação de classificador de matrizes para auxílio à escolha de estratégia de reordenação de matrizes

Matheus Alves da Silva, Thiago Gonçalves Mendes, Celmar Guimarães da Silva

Faculdade de Tecnologia, Universidade Estadual de Campinas

Resumo — Visualização de informação é uma área de pequisa que estuda técnicas para facilitar a compreensão de dados por meio do uso de gráficos interativos. Reordenar matrizes (como por exemplo as usadas para desenhar mapas de calor) pela permutação apropriada de suas linhas e colunas é uma técnica que possibilita evidenciar padrões ocultos em um conjunto de dados. A literatura define alguns tipos de padrões visuais de matrizes, ou seja, matrizes com características específicas que evidenciam determinados comportamentos globais ou locais em seu conjunto de dados. Este artigo apresenta, de forma resumida, um classificador baseado em técnicas de Aprendizagem de Máquina capaz de identificar o padrão subjacente a uma matriz não reordenada, dentre seis possíveis padrões: os 5 padrões canônicos de Wilkinson – Simplex, Band, Circumplex, Equi e Block – e um caso em que a matriz possui apenas ruídos. Com base em um vetor de características já identificado previamente na literatura, foram testados diferentes classificadores com diferentes hiperparâmetros, dentre os quais se escolheu um classificador baseado na técnica de Random Forest, cujos testes revelaram acurácia de 96%. O artigo também exemplifica o resultado da integração desse classificador a um método do estado da arte de reordenação de matrizes (Hybrid Sort).

Index Terms— Aprendizado de Máquina, Matrizes Reordenáveis, Visualização de Informação.

1 INTRODUÇÃO

Visualização de informação (InfoVis) é uma área da informática que estuda técnicas para facilitar a visualização do usuário em um conjunto grande e complexo de dados, utilizando recursos gráficos e contribuindo na busca e identificação de padrões. Ela é muito utilizada como auxílio para as áreas da sociologia, biologia, arqueologia, antropologia, cartografia, entre outros [5].

Dentre diversas técnicas para representação visual de dados, é de interesse para este projeto o conceito de *matriz reordenável* (Bertin, 2010). Trata-se de uma estrutura de dados que tem como foco possibilitar a permutação de linhas e colunas de uma matriz de dados sem perder a integridade do seu conteúdo, a fim de melhorar a visualização e interpretação de seus dados, permitindo o reconhecimento de padrões ao ser visualizada. A Figura 1 apresenta um exemplo de uma matriz de dados permutada, contendo dados sobre a semelhança entre conceitos relacionados ao humor das pessoas, conforme questionário preenchido por 472 pessoas [2].

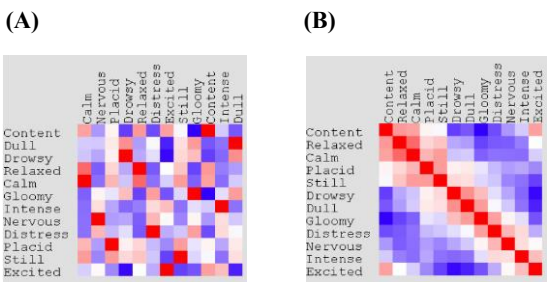

Figura 1 – Matriz sobre semelhança entre conceitos referentes a humor. Vermelho: alta semelhança, azul: baixa semelhança; branco: semelhança média. (a) Antes da permutação. (b) Após a permutação.

A literatura de InfoVis define alguns tipos de padrões de matrizes, ou seja, matrizes com características específicas que evidenciam determinados comportamentos globais ou locais no conjunto de dados por ela mostrado. Para estudos aprofundados sobre este tema, foi criada a ferramenta MRA (*Matrix Reordering Analyzer*) [7], cujo objetivo inicial é permitir fazer experimentos e comparar estaticamente algoritmos de reordenação de matrizes. Além disso, a ferramenta também permite realizar diferentes estudos de caso e identificar qual algoritmo de reordenação é o mais adequado para evidenciar, pela reordenação, um dos padrões de matriz. Atualmente, o grupo de pesquisa trabalha com cinco padrões propostos por Wilkinson – Simplex, Band, Circumplex, Equi-correlation (Equi) e Block [9].

Nesse contexto, esse projeto teve por objetivo a investigação de um classificador baseado em Aprendizagem de Máquina (*Machine Learning*) capaz de identificar o padrão de uma matriz não reordenada (ou seja, que não esteja propositadamente evidenciando algum padrão de matriz). O classificador faz essa classificação com base nos padrões de Wilkinson, citados anteriormente. O classificador elaborado foi integrado a um método do estado da arte de reordenação de matrizes (Hybrid Sort [8]), substituindo um classificador anterior que havia sido criado de maneira empírica.

Diferentes configurações de matrizes (tamanho, nível de ruído, variações de cada padrão) e de técnicas de Aprendizagem de Máquina foram utilizadas no processo, visando alcançar um resultado aceitável do ponto de vista científico. O resultado alcançado foi um classificador cuja avaliação retornou melhores resultados que o classificador empírico anterior, e ainda capaz de identificar matrizes que possuem apenas ruído.

O restante deste texto está organizado da seguinte forma: a Seção 2 aborda os trabalhos relacionados com este projeto; a Seção 3

apresenta os materiais e métodos utilizado nesta pesquisa, abordando a geração das amostras de matrizes e do modelo, o funcionamento do novo classificador e a implementação do algoritmo na ferramenta MRA; a Seção 4 apresenta os resultados finais obtidos neste projeto; e a Seção 5 apresenta a conclusão do projeto e aponta para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Utilizando os padrões de matriz de Wilkinson [9] (*canonical data patterns*), o grupo de pesquisa propôs anteriormente uma versão preliminar de um algoritmo híbrido (Hybrid Sort) [8]. Esse algoritmo inicialmente recebe uma matriz de dados a ser reordenada e a classifica como: pré-Simplex, pré-Equi, pré-Band, pré-Circumplex ou pré-Block. Uma matriz pré-Simplex é uma matriz para a qual há alguma permutação possível de linhas e/ou colunas, capaz de transformá-la em uma matriz Simplex. Os demais termos têm significado similar, e se aplicam aos demais padrões de matrizes.

Após a classificação em uma das 5 classes, é escolhido um algoritmo de reordenação correspondente à classe escolhida. Esse algoritmo é executado, reordenando a matriz de entrada e produzindo a saída final do Hybrid Sort, como ilustrado na Figura 2.

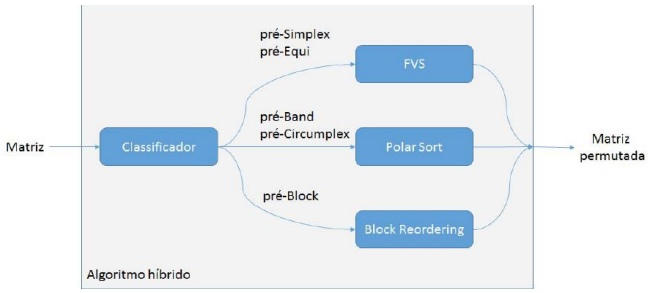

Figura 2 - Esquema de funcionamento do algoritmo híbrido.

O funcionamento desse algoritmo se baseia na geração de quatro vetores para análise: um vetor ordenado contendo os valores mínimos de cada linha da matriz; um vetor ordenado contendo os valores máximos de cada linha da matriz; um vetor ordenado contendo os valores mínimos de cada coluna da matriz; e um vetor ordenado contendo os valores máximo de cada coluna da matriz.

Usando esses vetores, é calculada uma regressão linear dos valores de cada vetor. Sendo $y=ax+b$ a expressão resultante de uma regressão linear, cada regressão feita gera dois coeficientes. Portanto, sendo 4 regressões, são gerados oito coeficientes, que formam um vetor de características da matriz: *aMinLinha*, *aMinCol*, *aMaxLinha*, *aMaxCol*, *bMinLinha*, *bMinCol*, *bMaxLinha* e *bMinCol*. Dentre eles, Medina (2014) havia experimentado usar os coeficientes angulares (os que se iniciam com a letra "*a*") como forma de classificar matrizes. Posteriormente, Silva verificou que é possível usar um subconjunto de 3 dos 8 coeficientes para a classificação, com resultados melhores que os de Medina [6]. Esses coeficientes delimitam uma região de valores no espaço $\Re^8$ para cada tipo de padrão de matriz, com valores específicos. Com base nessas regiões, foi criada uma árvore de decisão para a classificação dos padrões. Para cada matriz de entrada, o algoritmo calcula seu vetor de característica, e em seguida verifica em que região o vetor se localiza, classificando a matriz de entrada na classe relacionada a essa região.

Embora tenha produzido bons resultados de classificação, uma desvantagem do classificador usado é que ele foi construído empiricamente. Por isso, este trabalho focou-se em propor um classificador elaborado seguindo técnicas de classificação da área de Aprendizado de Máquina.

3 MATERIAIS E MÉTODOS

As seções a seguir detalham a metodologia para a criação do classificador, envolvendo as decisões para criação das amostras para treino de classificadores, a escolha de classificadores, o algoritmo criado e sua integração à ferramenta MRA.

3.1 GERAÇÃO DA AMOSTRA

Como preparação para o treino e escolha dos classificadores, foi necessária a geração de amostras de matrizes que posteriormente foram utilizadas para treinar o classificador, com diferentes padrões subjacentes, tamanhos e níveis de ruído.

Foram criadas matrizes contendo os cinco padrões propostos por Wilkinson – Band, Block, Circumplex, Equi e Simplex – como classes da amostra. Além disso, foi introduzida uma nova classe chamada "noise", visto que durante o andamento do projeto, notou-se a necessidade de diferenciar as matrizes que têm ruído igual ou superior a 90% em uma mesma classe, pois são matrizes muito ruidosas (ruídos, neste caso, são valores aleatórios inseridos em posições também aleatórias de uma matriz). Isso é necessário visto que com esse valor de ruído é muito difícil de se identificar e extrair informação visualmente (Figura 3), o que faria o classificador ser tendencioso. O nome dessa sexta classe foi inspirado em um antipadrão de matriz presente na literatura, chamado de "Noise Anti Pattern" [1]. Behrisch et al. definem esse antipadrão utilizando matrizes binárias; entretanto, em nosso projeto, ele foi adaptado para que seja o equivalente em matrizes não binárias.

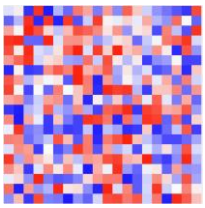

Figura 3 - Exemplo de matriz com 90% de ruído. Vermelho: valores altos; branco: valores intermediários; azul: valores baixos.

Foi decidido contemplar ordens de grandeza distintas, de forma similar a outros artigos propostos pelo grupo de pesquisa, para distribuição dos tamanhos e da quantidade de ruído das amostras. Foram geradas 120.000 amostras de matrizes para a construção do algoritmo, com a distribuição de 20.000 amostras para cada uma das seis classes definidas. A geração foi feita utilizando a ferramenta MRA.

As amostras tentaram representar matrizes de tamanho variado (tanto no número de colunas quanto no de linhas). Os valores utilizados para gerar as dimensões que essas amostras podem ter são 10, 12, 14, 17, 20, 24, 29, 35, 41, 49, 59, 70, 84, 100, 119, 143, 170, 203, 242, 289, 346, 412, 492, 588, 702, 838 e 1000, sendo que cada valor representa um número gerado para o tamanho da matriz, seja para o número de linhas ou o de colunas, de forma independente entre si. Estes valores foram gerados pela expressão $s=10k$, $k=1+i/13$, $i \in \{0,...,26\}$. Essa distribuição exponencial visa garantir uma quantidade igual de variações de tamanho por ordem de grandeza (dezenas e centenas de linhas e de colunas).

Para a geração dos níveis do ruído das amostras, foi utilizada uma distribuição similar, com a mesma técnica da geração dos tamanhos de matrizes, resultando nos níveis de ruídos 0.025, 0.032, 0.1, 0.126, 0.158, 0.2, 0.251, 0.316, 0.398, 0.501, 0.631, 0.794 e 1. Os níveis de ruídos foram definidos como $n=10k$, $k=-1{,}6+i/10$, $i \in \{0, \ldots, 16\}$.

Em relação à classe chamada “noise”, dentre o nível de ruído escolhidos em nossa lista de valores para ruídos, o único valor que representava mais de 90% de ruído foi 1, ou seja, 100% de ruído, tendo sido o valor que caracterizou esta nova classe de matrizes.

3.2 Geração do Modelo

Após a geração das amostras, a etapa seguinte envolve a construção de um modelo de classificação com o apoio da biblioteca WEKA [3], bastante conhecida pela implementação de algoritmos de Aprendizagem de Máquina. Além disso, essa biblioteca possui uma ramificação chamada de Auto-WEKA [4], cujo objetivo é encontrar o melhor algoritmo de classificação no contexto de Aprendizagem de Máquina e os melhores hiperparâmetros para as amostras fornecidas. Por este motivo, optou-se por utilizar a Auto-WEKA para a construção do modelo baseado no melhor algoritmo escolhido por ela.

Por fim, foi utilizada uma validação cruzada para a validação do classificador, utilizando o método k-fold (sendo k=10). O método consiste em dividir o conjunto de dados em k partes de mesmo tamanho e separando uma dessas partes para validação do modelo. O processo é repetido k vezes.

3.3 Integração do Modelo a Algoritmo de Reordenação de Matriz

Após a geração do modelo, a etapa seguinte inclui gerar uma nova versão do algoritmo Hybrid Sort, em que seu antigo classificador empírico seja substituído pelo novo classificador que é o modelo gerado na etapa anterior. Ao classificar uma matriz não reordenada em uma das 6 classes consideradas, o novo método, batizado de Hybrid Sort Plus, invoca o método de reordenação correspondente ao padrão da matriz classificado, como o Hybrid Sort fazia. Ao classificar uma matriz como “noise”, o Hybrid Sort Plus reordena a matriz de entrada nos três métodos de reordenação possíveis e escolhe o que tiver melhor resultado, de acordo com a avaliação das medidas de qualidade de ordenação de matrizes utilizadas pela ferramenta (Figura 4). Diferentes técnicas permitem essa avaliação, como por exemplo, a função de stress de Moore, Minimal Span Loss Function e a correlação circular de linhas e colunas.

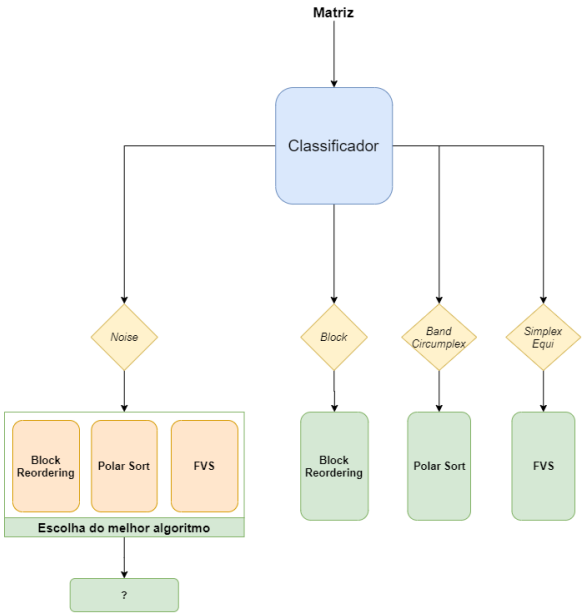

Figura 4: Funcionamento do algoritmo Hybrid Sort Plus.

4 Resultados

Os resultados apresentados nesta seção discorrem sobre a execução das etapas da pesquisa, sobre os resultados do classificador e uma breve comparação entre os classificadores do Hybrid Sort e do Hybrid Sort Plus.

As etapas de criação das amostras, geração do modelo e integração do modelo ao algoritmo Hybrid Sort foram efetuadas conforme planejado. Tendo sido definido o conjunto de amostras, foram preparados seus respectivos vetores de características para possibilitar a comparação das técnicas de classificação no contexto de Aprendizagem de Máquina. O Auto-WEKA foi utilizado como auxílio a esta etapa para comparação dos algoritmos, resultando na melhor técnica de classificação e os melhores hiperparâmetros dentre os testados. Este retornou o algoritmo Random Forest como o que produziu melhor classificação, com acurácia de 96%, como indicado na Tabela 1A. A Tabela 1B representa os resultados da matriz de confusão.

(A)

Correctly Classified Instances	**115.285 (96,071%)**
Incorrectly Classified Instances	**4.715 (3,929%)**
Kappa statistic	0.9529
Mean absolute error	0.023
Root mean squared error	0.1026
Relative absolute error	0,082793
Root relative squared error	0,2751
Total number of instances	120.000

(B)

	Band	Block	Circumplex	Equi	Noise	Simplex
Band	**98,10%**	0,20%	0,02%	0,09%	1,10%	0,50%
Block	0,79%	**98,79%**	0,00%	0,09%	0,24%	0,10%
Circumplex	0,03%	0,00%	**97,88%**	0,00%	2,10%	0,00%
Equi	0,03%	0,02%	0,00%	**98,36%**	1,41%	0,19%
Noise	0,17%	0,03%	0,68%	3,84%	**88,42%**	6,88%
Simplex	0,10%	0,00%	0,00%	0,10%	4,92%	**94,89%**

Tabela 1: Resultados da classificação do algoritmo. (A) Métricas estatísticas obtidas durante a classificação, geradas automaticamente pelo WEKA. (B) Resultados da matriz de confusão do Hybrid Sort Plus.

Dada uma matriz de entrada, o método que implementa o Hybrid Sort Plus, primeiramente calcula seu vetor de características. Em seguida, seu classificador (HybridSortPlus.model), gerado pelo Auto-WEKA, é carregado via serialização para dentro do método. Como terceiro passo, o vetor de características é repassado para o classificador, que efetua a classificação do vetor de características, retornando uma das seis classes de padrões.

Essa implementação foi feita com sucesso na ferramenta MRA. O algoritmo é colocado à disposição do usuário em uma caixa de seleção junto a outros algoritmos já implementados na MRA. Para a utilização do algoritmo, o usuário deve selecionar a matriz desejada e, posteriormente, escolher o algoritmo Hybrid Sort Plus para que ele classifique e execute o método de ordenação referente ao padrão identificado. A Figura 5 apresenta um exemplo de matriz reordenada pelo Hybrid Sort Plus na ferramenta MRA.

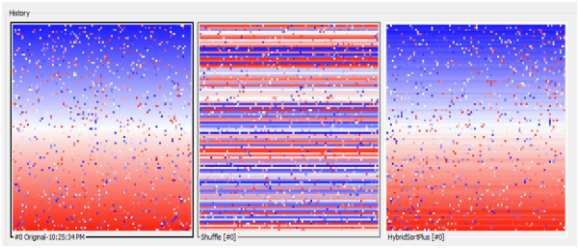

Figura 5: Matriz 100 × 100 com o padrão Equi com 10% de ruído, reordenada pelo algoritmo Hybrid Sort Plus implementado na ferramenta MRA. Da esquerda para direita: a matriz original, a mesma matriz com os dados embaralhados e posteriormente reordenada pelo Hybrid Sort Plus.

Para comparar os classificadores dos algoritmos Hybrid Sort e Hybrid Sort Plus, foi realizado um experimento utilizando 90.000 amostras geradas de forma aleatória. Gerando bons resultados, o classificador do Hybrid Sort Plus obteve 67.420 amostras classificadas corretamente contra 55.365 obtidas pelo classificador do Hybrid Sort. Em relação aos erros, o classificador do Hybrid Sort Plus obteve somente 1.164 amostras classificadas incorretamente contra 13.219 obtidas pelo classificador antigo (Figura 6).

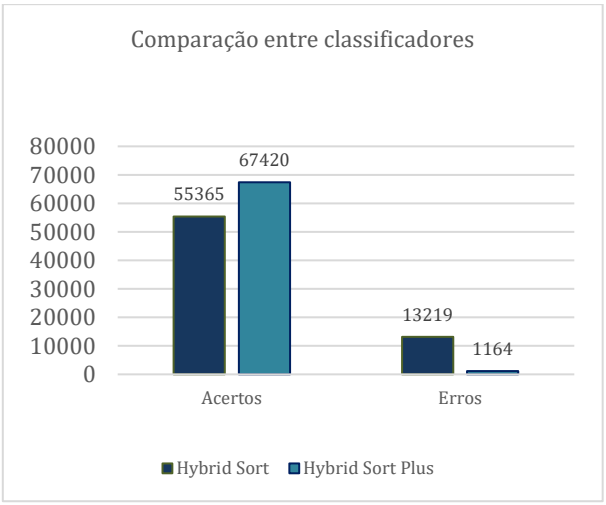

Figura 6: Comparação dos algoritmos Hybrid Sort e Hybrid Sort Plus.

Além disso, foi gerado a matriz de confusão dos algoritmos Hybrid Sort (Tabela 2A) e Hybrid Sort Plus (Tabela 2B) para uma comparação mais aprofundada.

(A)

	Band	Block	Circumplex	Equi	Simplex
Band	**86,49%**	0,00%	9,93%	0,18%	3,39%
Block	0,67%	**85,95%**	12,96%	0,42%	0,00%
Circumplex	2,56%	0,00%	**96,73%**	0,00%	0,70%
Equi	7,05%	0,00%	0,07%	**61,66%**	31,22%
Simplex	23,08%	0,00%	3,46%	0,20%	**73,26%**

(B)

	Band	Block	Circumplex	Equi	Noise	Simplex
Band	**97,09%**	2,69%	0,01%	0,02%	0,13%	0,07%
Block	0,00%	**100,00%**	0,00%	0,00%	0,00%	0,00%
Circumplex	0,03%	0,00%	**99,75%**	0,00%	0,22%	0,00%
Equi	0,02%	0,01%	0,00%	**99,04%**	0,90%	0,03%
Noise	0,00%	0,00%	0,00%	0,00%	**0,00%**	0,00%
Simplex	0,17%	0,01%	0,00%	0,04%	4,09%	**95,68%**

Tabela 2: Comparação das matrizes de confusão dos algoritmos. (A) Hybrid Sort, algoritmo gerado de forma empírica. (B) Hybrid Sort Plus, algoritmo gerado com técnicas de Aprendizagem de Máquina.

Por fim, foi gerado os resultados das métricas de classificação dos algoritmos Hybrid Sort (Tabela 3A) e Hybrid Sort Plus (Tabela 3B).

(A)

Accuracy score	0,8073
Precision score	0,8332
Recall score	0,8073
F1 score	0,8071

(B)

Accuracy score	0,9830
Precision score	0,9939
Recall score	0,9830
F1 score	0,9883

Tabela 3: Comparação das métricas estatísticas dos algoritmos. (A) Métricas estatísticas do algoritmo Hybrid Sort (B) Métricas estatísticas do algoritmo Hybrid Sort Plus.

Estes resultados são um bom indício de que o novo classificador supera o antigo, visto que o Hybrid Sort Plus apresentou resultados altamente superiores ao Hybrid Sort.

5 Conclusão

O algoritmo de reordenação Hybrid Sort carecia de um classificador que superasse os resultados do classificador empírico previamente desenvolvido. O novo classificador proposto alcançou esses objetivos, provendo uma classificação mais precisa das matrizes que possuem padrões de Wilkinson subjacentes a elas, e ainda diferenciando parcialmente as matrizes ruidosas daquelas que realmente possuem padrões. Esse classificador foi inserido no método de reordenação, que passou a ser chamado de Hybrid Sort Plus. Comparações iniciais indicaram ótimos resultados de matrizes reordenadas, uma vez que as classificações das matrizes de entrada estão melhores.

Trabalhos futuros referentes a este tema incluem ampliar o classificador gerado neste projeto para novos padrões de matrizes (Bands, Block, Off-diagonal block, e Line/star), propostos por Behrisch et al. [1], bem como estudar algoritmos de reordenação capazes de evidenciar esses novos padrões e que possam ser adicionados ao Hybrid Sort Plus.

Agradecimentos

Agradecemos ao SAE (Serviço de Apoio ao Estudante da Universidade Estadual de Campinas) que financiou a bolsa de Iniciação Científica de M. A. Silva. Agradecemos também ao Prof. Dr. Guilherme Palermo Coelho pelas colaborações no tema de Aprendizado de Máquina, essenciais para o desenvolvimento deste projeto.

Referências

[1] BEHRISCH, M. BACH, B., HENRY-RICHE, N., SCHRECK, T., FEKETE, J.-D. Matrix Reordering Methods for Table and Network Visualization. Computer Graphics Forum 35 (3), 2016, pp. 693-716.

[2] BROWNE, M.W. Circumplex Models for Correlation Matrices. Psychometrika 57(4), 1992, pp. 469-497.

[3] HALL, Mark et al. The WEKA data mining software. ACM SIGKDD Explorations Newsletter, v. 11, n. 1, p. 10–18, 2009.

[4] KOTTHOFF, Lars et al. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. Journal of Machine Learning Research, v. 18, p. 1–5, 2017.

[5] LIIV, I. Seriation and Matrix Reordering: An Historical Overview. Wiley InterScience, 2010.
[6] MEDINA, B.F. Reordenação de matrizes de dados quantitativos usando árvores PQR. 2015. 77 f. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Tecnologia, Limeira, SP.
[7] SILVA, C. G., MELO, M. F., SILVA, F. P., MEIDANIS, J. PQR Sort - Using PQR trees for binary matrix for binary matrix reorganization. Journal of the Brazilian Computer Society. 2013. 10.1186/1678-4804-20-3.
[8] SILVA, C. G.. Hybrid Sort - A pattern-focused matrix reordering approach based on classification. In: 14th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, 2020, (online). Proceedings of the International Conferences on Computer Graphics, Visualization, Computer Vision and Image Processing 2020, Big Data Analytics, Data Mining and Computational Intelligence 2020, and Theory and Practice in Modern Computing 2020, 2020. p. 35-43.
[9] WILKINSON, L. The Grammar of Graphics. 2. ed. [S. l.]: Springer, 2005. 693 p.