AutoCAR: Automação e Reprodutibilidade de Testes de Métodos de Classificação Baseados em Regras de Associação

Vanderson da Silva Rocha[1], **Diego Kreutz**[2], **Eduardo Feitosa**[1]

[1]Universidade Federal do Amazonas (UFAM)
vanderson@ufam.edu.br,efeitosa@icomp.ufam.edu.br

[2]Universidade Federal do Pampa (UNIPAMPA)
DiegoKreutz@unipampa.edu.br

Resumo. *Existem diferentes métodos de classificação baseados em regras de associação. Entretanto, é muitas vezes difícil de encontrar a implementação e realizar a avaliação dos métodos. Para mitigar esses problemas, propomos a AutoCAR, um arcabouço de software modular para catalogar implementações, automatizar os testes e a avaliação de métodos de classificação baseados em regras de associação. AutoCAR permite que novos métodos sejam incorporados à ferramenta sem a necessidade de modificação do código.*

1. Introdução

Existem três algoritmos clássicos de mineração por regras de associação, o Apriori, FP-Growth, e ECLAT [Li and Sheu, 2021]. A partir deles surgiram diversos métodos especializados de classificação baseados em regras de associação, definidos como integração de classificação e mineração por regras de associação [Liu et al., 1998], como CBA [Liu et al., 1998], CMAR [Li et al., 2001], CPAR [Yin and Han, 2003], ARCID [Abdellatif et al., 2018] e EQAR [Rocha et al., 2022], que incorporam e evoluem os algoritmos clássicos.

Na literatura existem trabalhos que propõe e avaliam métodos de classificação baseados em regras de associação para diferentes domínios e contextos, como saúde e Big Data [Wang et al., 2012, Mattiev and Kavsek, 2020, Li and Sheu, 2021]. Alguns trabalhos realizam análises pontuais de desempenho de métodos como o CBA [Jirı and Kliegr, 2014], análises de cobertura de diferentes métodos (e.g., CBA, CPAR, CMAR) [Mattiev and Kavsek, 2020], ou ainda a proposição de novos métodos (e.g., WCBA) e comparação com alguns dos existentes [Alwidian et al., 2018]. Entretanto, é importante destacar que nenhum dos trabalhos encontrados disponibiliza software (e.g., uma ferramenta ou conjuntos de implementações) para reprodução ou expansão das avaliações realizadas, o que dificulta a reprodutibilidade e a construção e comparação contínua de novas propostas.

Com o objetivo de avaliar diferentes métodos de classificação baseados em regras de associação, no contexto de detecção de *malwares* Android, e também facilitar a reprodutibilidade e expansão contínua das pesquisas na temática, propomos a ferramenta denominada AutoCAR. Em sua essência, a ferramenta consiste de um arcabouço de software modular, que permite a incorporação das mais diversas implementações de métodos de classificação baseados em regras de associação e a avaliação sistematizada e simplificada dos métodos. Como ponto de partida e demonstração, apresentamos a incorporação de quatro métodos de classificação baseados em regras de associação (CBA, CMAR, CPAR,

EQAR), escritos em diferentes linguagens de programação, e dois modelos de aprendizado de máquina, o *Random Forest* (RF) e o *Support Vector Machine* (SVM). A AutoCAR é capaz de executar e comparar, apresentando gráficos visuais, um número qualquer de métodos de classificação e modelos de aprendizado de máquina, facilitando a reprodutibilidade e a evolução de trabalhos baseados nesse tipo de métodos.

O restante deste trabalho está organizado como segue. Nas Seções 2 e 3 apresentamos a modelagem, a implementação e uma avaliação da AutoCAR. Por fim, na Seção 4 apresentamos as informações básicas sobre a demonstração e as considerações finais.

2. A ferramenta AutoCAR

Nesta seção detalhamos a modelagem do fluxo de execução (Seção 2.1) e a implementação (Seção 2.2) da AutoCAR. Como poderá ser observado, a ferramenta é composta por três módulos principais: entrada de dados, execução dos métodos ou modelos e geração dos gráficos para comparação dos modelos.

2.1. Modelagem

Podemos observar os três módulos do fluxo de execução da ferramenta AutoCAR na Figura 1. No primeiro módulo, de **Dados de Entrada**, a ferramenta recebe e realiza a validação dos parâmetros de entrada, analisando as opções indicadas pelo usuário (e.g., execução do modelo A ou de todos os modelos), os limites (e.g., valores mínimos e máximos para parâmetros de qualidade), os tipo de dados (e.g., inteiro, *string*) e os requisitos obrigatórios (e.g., parâmetro de qualidade para modelos de classificação baseados em regras de associação). Adicionalmente, neste módulo, como parte do fluxo de execução, é realizado o carregamento e pré-processamento (e.g., identificação de inconsistências) dos *datasets* (i.e., conjuntos de dados) que serão disponibilizados para o módulo de **Execução dos Modelos**.

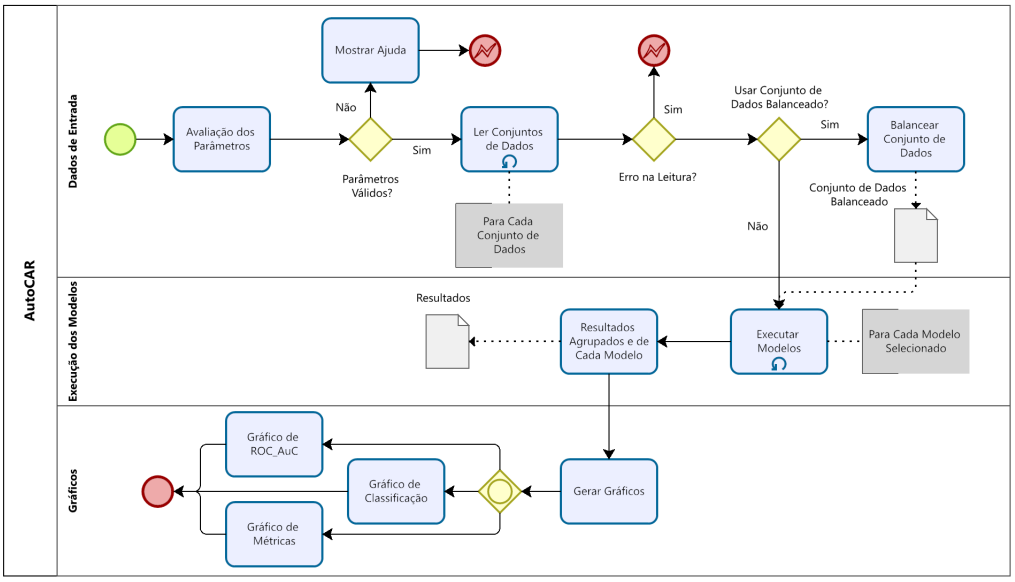


Figura 1. Modelagem do fluxo de execução da AutoCAR

No módulo de **Execução dos Modelos** são instanciados todos os métodos de classificação baseados em regras de associação e modelos de aprendizado de máquina

selecionados pelo usuário. Cada método e modelo é executado para cada um dos conjuntos de dados de entrada informados pelo usuário. Para tanto, utilizamos a técnica de validação cruzada estratificada, dividindo e alternando os conjuntos de treino e teste. O número de partições (*folds*) foi fixado em 5 ($k = 5$), o que significa que em cada iteração 80% das amostras serão utilizadas para treino e 20% para teste. O resultado de cada execução é salvo individualmente. Na sequência, ocorre também um agrupamento dos resultados para a geração automática de gráficos comparativos.

No último módulo, na configuração padrão, a AutoCAR gera três tipos de **Gráficos**: (a) **de métricas**, utilizando barras agrupadas para as métricas de avaliação de cada conjunto de dados, para cada um dos métodos ou modelos selecionados; (b) **de classificação**, utilizando barras empilhadas para exibir o percentual de distribuição da classificação das amostras (e.g., falsos positivos); e (c) **de ROC_AuC**[1], utilizando dispersão para demonstrar os valores da métrica ROC_AuC obtidos pelos métodos e modelos de classificação.

2.2. Implementação

Na Figura 2 apresentamos uma visão geral da implementação da AutoCAR, indicando os parâmetros de entrada, as ferramentas e tecnologias utilizadas e as saídas.

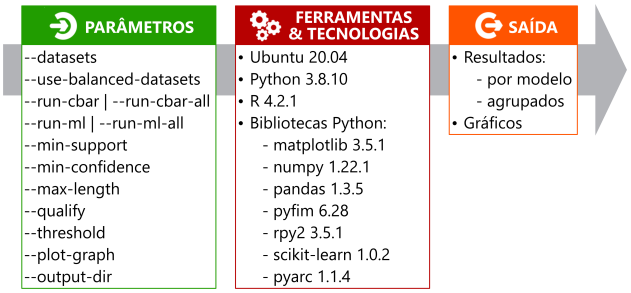


Figura 2. Implementação da AutoCAR

A Tabela 1 resume os parâmetros de entrada disponíveis na ferramenta. Parâmetros como o `--use-balanced-datasets` permitem o balanceamento automático de *datasets* desbalanceados, que podem enviesar o aprendizado dos modelos. Já parâmetros de entrada como `--run-cbar` e `--run-ml` permitem escolher os métodos e modelos que serão executados (e.g., `--run-cbar-all` ou `--run-cbar cba`). Por fim, parâmetros como `--min-support`, `--min-confidence`, `--max-length`, `--qualify` e `--threshold` afetam a execução dos métodos propriamente ditos. Os valores padrão estão definidos de acordo com recomendações da literatura.

Para a implementação e automação da execução do AutoCAR, utilizamos a linguagem Python (versão 3.8.10) e as bibliotecas NumPy (versão 1.22.1), Pandas (versão 1.3.5), scikit-learn (versão 1.0.2) e Matplotlib (versão 3.5.1). A Tabela 2 apresenta também as linguagens e bibliotecas de implementação dos métodos de classificação atualmente disponíveis na ferramenta. Os métodos CMAR e o CPAR, implementados em linguagem R, foram integrados na ferramenta através da biblioteca rpy2[2] (versão 3.5.2).

[1] `https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5`

[2] `https://pypi.org/project/rpy2/`

Tabela 1. Parâmetros de entrada da ferramenta

Parâmetro	Descrição
`--datasets`	Especifica a lista dos *datasets* a serem avaliados.
`--use-balanced-datasets`	Gera o balanceamento automático dos *datasets*.
`--run-cbar` | `--run-cbar-all`	Permite selecionar os métodos de classificação.
`--run-ml` | `--run-ml-all`	Permite selecionar os modelos de aprendizado de máquina.
`--min-support`	Porcentagem (valor padrão 0.1) de transações que devem conter a união dos itens da regra.
`--min-confidence`	Probabilidade (valor padrão 0.95) de ocorrer o consequente Y em uma transação que contém o antecedente X.
`--max-length`	Número máximo de itens em cada regra (valor padrão 5).
`--qualify`	Métrica a ser utilizada para qualificar as regras.
`--threshold`	Porcentagem (valor padrão 0.2) de regras a serem utilizada para avaliar as amostras.
`--plot-graph`	Habilita a geração de gráficos para as métricas resultantes dos métodos e modelos.

Para permitir a fácil e rápida integração de outros modelos à ferramenta, utilizamos uma estrutura de diretórios e arquivos similar às bibliotecas utilizadas no `gcc` do Linux. Por exemplo, a inclusão de um novo modelo requer apenas um novo sub-diretório dentro do diretório `models` e um arquivo padrão de invocação (i.e., `run.py`), cuja função `run` deve receber como argumentos de entrada o *dataset* e os parâmetros da ferramenta (e.g., prefixo dos arquivos de saída). Em cada sub-diretório, podem ser adicionados também arquivos como o `about.desc`, que descreve o novo modelo para a AutoCAR. Atendidos esses requisitos mínimos, o novo método ou modelo passa automaticamente a estar disponível, como novo parâmetro de execução, na ferramenta.

Tabela 2. Implementação do Modelos

Modelo	Linguagem	Bibliotecas/Pacotes
CBA	Python (versão 3.8.10)	pyARC[3] (versão 1.1.4)
CMAR	R (versão 4.2.1)	arulesCBA[4] (versão 1.2.4)
CPAR	R (versão 4.2.1)	arulesCBA (versão 1.2.4)
EQAR	Python (versão 3.8.10)	pyFIM[5] (versão 6.28)
RF	Python (versão 3.8.10)	scikit-learn (versão 1.0.2)
SVM	Python (versão 3.8.10)	scikit-learn (versão 1.0.2)

3. Avaliação

Nesta seção apresentamos um resumo dos métodos de classificação baseados em regras de associação e modelos de aprendizado de máquina disponíveis no arcabouço da AutoCAR (Seção 3.1). Na sequência, descrevemos os *datasets* (Seção 3.2) e os parâmetros

[3] `https://github.com/jirifilip/pyARC`
[4] `https://github.com/ianjjohnson/arulesCBA`
[5] `https://borgelt.net/pyfim.html`

e métricas dos modelos (Seção 3.3) utilizados na execução. Finalmente, apresentamos o ambiente onde os experimentos foram executados (Seção 3.4) e os resultados (Seção 3.5).

3.1. Métodos e Modelos Classificação

Na Tabela 3 apresentamos um resumo das principais semelhanças e diferentes entre os quatro métodos de classificação baseados em regras de associação (CBA, CMAR, CPAR, EQAR) disponível no arcabouço da AutoCAR. Como podemos observar, há uma diversidade de algoritmos de mineração utilizados pelos métodos para geração das regras, incluindo os três clássicos (i.e., Apriori/CBA, FP-Growth/CMAR e ECLAT/EQAR) e o PRM [Yin and Han, 2003], empregado pelo CPAR. Para classificação de regras, CBA, CMAR e EQAR utilizam essencialmente confiança e suporte. Já o CPAR emprega a acurácia de Laplace para estimar o erro esperado de uma regra.

Tabela 3. Métodos de Classificação baseados em Regras de Associação

Método	Geração de Regras	Classificação das Regras	Poda	Predição
CBA	Apriori	Confiança, Suporte, Geradas Primeiro	Cobertura (Método M1)	Máxima Probabilidade
CMAR	FP-Growth	Confiança, Suporte, Cardinalidade	Cobertura (Método M1), Regras Redundantes	Múltiplas Regras
CPAR	PRM	Acurácia de Laplace	Seleção dos K Melhores	Múltiplas Regras
EQAR	ECLAT	Confiança e Suporte	Qualidade de Regra, Diferença de Conjuntos	Correspondência Exata com Alguma Regra

3.2. *Datasets*

Para avaliar e comparar os métodos e modelos disponíveis na AutoCAR, utilizamos como base seis *datasets* disponíveis publicamente: duas instâncias do KronoDroid, DREBIN-215, AndroCrawl, ANDROIT e DefenseDroid, conforme detalhado na Tabela 4. É importante destacar que esses *datasets* são utilizados com frequência por diversos trabalhos atuais de detecção de *malwares* Android (e.g., [Guerra-Manzanares et al., 2021]).

Tabela 4. *Datasets*

Dataset	Características	Amostras		
		Maliciosos	Benignos	Total
AndroCrawl	81	10170	86562	96732
ANDROIT	166	3418	8058	11476
DefenseDroid	2938	6000	5975	11975
DREBIN-215	215	5560	9476	15036
KronoDroid Dispositivo Real	383	41382	36755	78137
KronoDroid Emulador	383	28745	35246	63991

3.3. Parâmetros e Métricas

A Tabela 5 resume os principais parâmetros e valores utilizados na avaliação. O valor de suporte desempenha um papel importante na previsão geral de classificadores derivados de técnicas de classificação baseadas em regras de associação existentes. Se utilizarmos um valor de suporte muito alto, corremos o risco de ignorar regras de boa qualidade. Por outro lado, com um suporte muito baixo, potencializamos o problema de *overfitting*, levando a regras redundantes que irão consumir mais tempo de processamento.

Tabela 5. Parâmetros

Parâmetro	Valores Utilizados
`--min-support`	10%
`--min-confidence`	0.95
`--max-length`	5
`--qualify`	prec
`--threshold`	10%

A partir de observações na literatura e experimentos inicias, fixamos o **tamanho máximo da regra** em **5** para geração das regras. Fixamos ainda a **confiança** em 95%, um valor adequado para selecionarmos as características que ocorrem simultaneamente na maioria das amostras dos *datasets*.

Com relação aos modelos de aprendizado de máquina (RF e SVM), utilizamos a execução padrão apresentada na biblioteca *scikit-learn*, sem qualquer alteração de parâmetro ou hiper-parâmetro. O RF é executado com 100 árvores na floresta e considerando a raiz quadrada da quantidade de características ao procurar a melhor divisão entre as classes. Já o SVM utiliza o kernel *Radial Basis Function* (RBF) e heurística de redução para diminuir o tempo de treino no caso de muitas interações.

Como métricas de avaliação, além das tradicionais (acurácia, precisão, *recall* e F1 *Score*), utilizamos também o Coeficiente de Correlação de Matthews (MCC) [Chicco and Jurman, 2020], que é particularmente útil quando as duas classes (maliciosos e benignos) estão desequilibradas, ou seja, uma classe aparece muito mais que a outra.

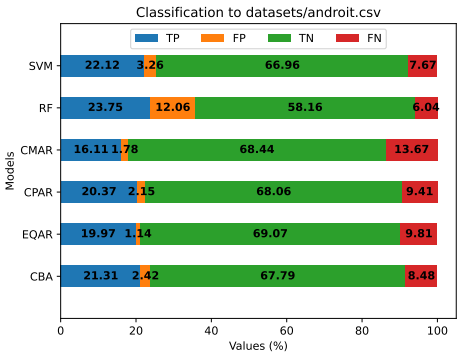
3.4. Ambiente

Para a execução dos experimentos, utilizamos um computador com processador Intel Xeon E5-5649 six-core de 2.53GHz, com 96GB RAM e 150GB de HD. O sistema operacional utilizado foi o Linux Ubuntu Server 20.04.3 LTS.

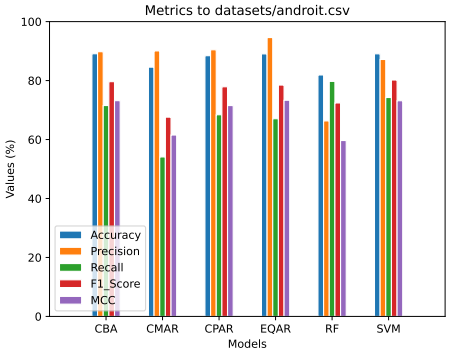
3.5. Resultados

A Figura 3 apresenta os três gráficos padrão gerados automaticamente pela AutoCAR, incluído classificação, métricas e ROC_AuC. Os gráficos da figura representam os resultados para o *dataset* ANDROIT. Os demais gráficos, para os outros *datasets*, estão disponíveis no repositório da ferramenta[6]. Observando o gráfico de classificação da Figura 3a, podemos constatar que o RF obteve aproximadamente 12% de falsos positivos, enquanto a maior taxa de falsos negativos foi do modelo CMAR.

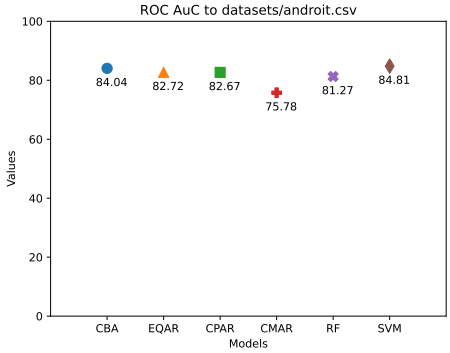
[6]`https://github.com/Malware-Hunter/sf22-autoCAR`



(a) Gráfico de Classificação



(b) Gráfico de Métricas



(c) Gráfico de ROC AuC

Figura 3. Gráficos resultantes da AutoCAR

No gráfico de métricas (Figura 3b), podemos observar que os modelos que utilizam aprendizado de máquina (i.e., RF e SVM) apresentaram variação de métricas semelhantes aos métodos de classificação baseados em regras de associação. Os números indicam que o RF e o CMAR apresentaram os menores valores de MCC e *recall*, respectivamente. Já no gráfico de ROC_AuC (Figura 3c), podemos observar que o método CMAR possui a menor desempenho geral de classificação. É importante destacar que esses gráficos comparativos, gerados automaticamente pela AutoCAR, simplificam e agilizam a interpretabilidade dos dados.

4. Considerações Finais

Demonstração

O código-fonte, documentação, instruções de utilização e instalação nativa ou via Docker estão dispoıveis no repositório da AutoCAR[7]. A demonstração da ferramenta será realizada através de um ambiente virtual, hospedado em dispositivo próprio dos autores. O funcionamento da ferramenta será demonstrado através dos seguintes passos: (a) apresentação dos parâmetros de execução da ferramenta; (b) apresentação da estrutura de diretórios utilizada; (c) apresentação das funcionalidades da ferramenta; (d) detalhamento e demonstração da inclusão de um novo modelo na ferramenta; e (e) apresentação dos resultados (i.e., gráficos) gerados pela AutoCAR.

[7]`https://github.com/Malware-Hunter/sf22-autoCAR`

Conclusão

Nós apresentamos a ferramenta AutoCAR, um arcabouço de software, modular, que permite a inclusão simplificada de novos métodos e modelos de classificação, bem como sua avaliação de forma completamente automatizada, potencializando significativamente a reprodutibilidade e a avaliação de novos métodos de classificação. É importante destacar que a ferramenta facilita também a análise rápida dos métodos e modelos através de métricas apresentadas em gráficos, uma das saídas da ferramenta. Enfim, acreditamos que o AutoCAR é um primeiro passo para melhorarmos a reprodutibilidade e a evolução de pesquisas que envolvem métodos de classificação baseados em regras de associação.

Como trabalhos futuros, podemos destacar: (a) inclusão de novos métodos de classificação baseados em regras de associação; (b) inclusão de novos modelos de aprendizado de máquina; (c) ampliação do número de *datasets* disponíveis, fornecendo uma maior diversidade e robustez para as avaliações; (d) incorporação de níveis avançados e mais detalhados, configuráveis pelo usuário, de acompanhamento da execução e depuração da ferramenta.

Agradecimentos. Esta pesquisa foi parcialmente financiada, conforme previsto nos Arts. 21 e 22 do decreto no. 10.521/2020, nos termos da Lei Federal no. 8.387/1991, através do convênio no. 003/2021, firmado entre ICOMP/UFAM, Flextronics da Amazônia Ltda e Motorola Mobility Comércio de Produtos Eletrônicos Ltda. O presente trabalho foi realizado também com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

Abdellatif, S., Ben Hassine, M. A., Ben Yahia, S., and Bouzeghoub, A. (2018). ARCID: a new approach to deal with imbalanced datasets classification. In *Int. Conf. on Cur. Trends in Th. and Pr. of Inf.*, page 569.

Alwidian, J., Hammo, B. H., and Obeid, N. (2018). WCBA: Weighted classification based on association rules algorithm for breast cancer disease. *Applied Soft Computing*, 62:536–549.

Chicco, D. and Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13.

Guerra-Manzanares, A., Bahsi, H., and Nõmm, S. (2021). Kronodroid: Time-based hybrid-featured dataset for effective android malware detection and characterization. *Computers & Security*, 110:102399.

Jirı, F. and Kliegr, T. (2014). Classification based on associations (cba)-a performance analysis. In *CEUR*.

Li, H. and Sheu, P. C.-Y. (2021). A scalable association rule learning heuristic for large datasets. *Journal of Big Data*, 8(1):1–32.

Li, W., Han, J., and Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings 2001 IEEE international conference on data mining*, pages 369–376.

Liu, B., Hsu, W., Ma, Y., et al. (1998). Integrating classification and association rule mining. In *Kdd*, volume 98, pages 80–86.

Mattiev, J. and Kavsek, B. (2020). Coverage-based classification using association rule mining. *Applied Sciences*, 10(20).

Rocha, V. d. S., Kreutz, D., Feitosa, E. L., and Pontes, J. (2022). Avaliação de métodos de classificação baseados em regras de associação para detecção de malwares android. In *SBSeg22*, pages 1–14. SBC.

Wang, M., Su, X., Liu, F., and Cai, R. (2012). A cancer classification method based on association rules. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1094–1098.

Yin, X. and Han, J. (2003). CPAR: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 331–335. SIAM.