
Classificação *Online*, Evolutiva, Multi-rótulos, Baseada em Regras *Fuzzy*, Teoria dos Valores Extremos e Aprendizado de Máquina

Palavras-chave: Aprendizado Multitarefa, Sistemas Evolutivos *Fuzzy*, Classificação Multi-rótulos

Autores/as:

Maria Júlia De Castro Villafranca Garcia - FEEC/UNICAMP

Prof^o. Dr^o. Fernando José Von Zuben (orientador) - FEEC/UNICAMP

Dr^a. Amanda Ortega De Castro Ayres (coorientadora) - FEEC/UNICAMP

1. Introdução

Dados em tempo-real, ou fluxo de dados, podem ser definidos como uma sequência de atributos ordenada, rápida, contínua, potencialmente infinita e suscetível a mudanças dinâmicas. A crescente produção destes tipos de dados em diversos formatos e dispositivos de comunicação deve ser processada, organizada, e classificada de modo a atender eficientemente determinados fins como tomada de decisões econômicas ou corporativas, recomendações de filmes, vídeos e produtos em plataformas online, entre outros. A automatização da categorização e rotulação de toda essa informação se mostra imprescindível não só pela quantidade de dados a ser processada, mas também pela importância de uma criteriosa manipulação. Ademais, é comum que, durante a rotulação, um determinado dado, como uma foto, uma música ou um texto publicado na internet, possa pertencer a mais de uma categoria ao mesmo tempo. Dessa forma, há a necessidade de algoritmos capazes de classificar os dados multi-rótulos em fluxo de dados *online* [1].

Para suprir essa necessidade, esta pesquisa propõe a adaptação de um algoritmo de predição de séries temporais para um algoritmo de classificação multi-rótulos em fluxo de dados *online*. Os algoritmos adaptados de [2], [3] e chamados de Extreme Value evolving Classifier (EVeC) baseiam-se em regras *fuzzy* do tipo Takagi-Sugeno e possuem a função de pertinência dos antecedentes e consequentes das regras descritas pela distribuição Weibull. Cada uma das versões desenvolvidas do algoritmo utilizou uma abordagem diferente na saída dos consequentes das regras e na estimativa de seus parâmetros, de modo a evidenciar a vantagem da inserção do aprendizado multitarefa na predição dos rótulos para um determinado conjunto de instâncias de entrada. A versão final, que emprega aprendizado multitarefa, mostrou um bom desempenho frente às versões anteriormente analisadas, apresentando também um desempenho competitivo frente a outros algoritmos de classificação multi-rótulos em fluxo de dados da literatura.

2. Metodologia

2.1. Aprendizado de Máquina e Sistemas Evolutivos *Fuzzy*

Quando a predição de rótulos para um conjunto de instâncias é feita sobre dados no modo *offline*, é na etapa de treinamento do aprendizado de máquina que o algoritmo recebe um conjunto de amostras rotuladas, para então ser capaz de rotular corretamente instâncias não rotuladas e desconhecidas na etapa de teste [6]. Entretanto, em uma situação de classificação *online*, os dados não são conhecidos a priori, o que inviabiliza, em princípio, a etapa de treinamento. Para lidar com essa dificuldade, surgiram os algoritmos evolutivos (do inglês, *evolving*), que são capazes de adaptar sua

estrutura e seus parâmetros *online* de modo a oferecer previsões coerentes para dados que chegam continuamente ao longo do tempo. Um dos tipos mais conhecidos de sistemas evolutivos são os sistemas evolutivos *fuzzy*, baseados em regras fuzzy, que empregam regras que são continuamente revistas conforme a necessidade da distribuição dos dados [4].

2.1.1. Sistemas Evolutivos *Fuzzy*

Conjunto de dados *fuzzy* podem ser vistos como uma extensão da teoria de conjuntos clássica em que um elemento pode assumir apenas dois valores extremos, 0 ou 1. Para tanto, os conjuntos *fuzzy* são definidos por funções de pertinência contínuas que possuem associados graus de pertinência, compreendidos no intervalo [0, 1]. Para as funções de pertinência, adotou-se neste projeto a distribuição Weibull, que é um forma mais generalizada do que as geralmente utilizadas na literatura - triangulares, trapezoidais, ou em forma de sino (gaussianas) [4]. Ademais, dentre diversos modelos de inferência para sistemas baseados em regras fuzzy, o modelo Takagi-Sugeno (TS) é o adotado no projeto. Ao contrário do que acontece no modelo Mamdani, no modelo TS tem-se os consequentes das regras sendo formados por funções não *fuzzy* (funções *crisp*). Em um primeiro momento, a forma mais simples de construção das regras TS pode ser representada da seguinte maneira:

$$R^i : SE (x_1 \text{ está próximo de } x_{0_1}^i) \dots E (x_j \text{ está próximo de } x_{0_j}^i) \dots E (x_n \text{ está próximo de } x_{0_n}^i) \\ ENTÃO \hat{y}^i \quad (1)$$

onde R^i corresponde à i -ésima regra do modelo, de modo que $i = 1, 2, \dots, c$, c é o número de regras, $x = [x_1, \dots, x_j, \dots, x_n]^T \in R^n$ é o vetor de variáveis de entrada, x_0^i é o ponto focal do antecedente da i -ésima regra e \hat{y}^i é a saída predita para a regra R^i .

2.2. Extreme Value evolving Classifier (EVeC)

O algoritmo desenvolvido nesta pesquisa, chamado Extreme Value evolving Classifier (EVeC), é capaz de aprender *from scratch*, uma vez que novas regras podem ser formadas, fundidas ou deletadas a partir do contínuo fluxo de dados que chega. Para tanto, no início do programa, hiperparâmetros, a definir, auxiliam na auto-adaptação da estrutura e dos parâmetros do modelo. A saber, os hiperparâmetros a definir são: σ : refere-se à granularidade, N : refere-se à quantidade de pontos de dados alocados para cada regra do sistema de janelamento: δ : refere-se ao período de atualização do modelo; e somente para a versão que emprega o aprendizado multitarefa, ρ : usado como peso no termo de regularização.

1. Classificação multi-rótulos sem regressão

Na versão 0, em que o consequente das regras é simplesmente um rótulo, o algoritmo percorre todas as regras e verifica se o grau de ativação da amostra para aquela regra é maior que ou igual ao limiar σ . A saída do algoritmo conterá 1 para todos os rótulos cujo grau de ativação para as respectivas regras foi maior que ou igual a σ e 0, caso contrário. Para esta versão, as regras do modelo de Takagi-Sugeno (TS), ficam da seguinte forma:

$$\hat{L}^i = \hat{y}^i = \begin{cases} 1, & \text{se } \mu^i \geq \sigma \\ 0, & \text{caso contrário} \end{cases}$$

2. Classificação multi-rótulos com regressão

Outra forma de fazer a defuzzificação clássica da saída das regras é através da média ponderada entre as saídas locais preditas por cada uma das regras e normalizadas, ponderadas pela função de pertinência referente à regra em questão. Ademais, nesta versão os parâmetros dos consequentes das regras são estimados por quadrados mínimos [5]. O rótulo final de cada classe será então 1, se a saída da classe for maior que ou igual a um *threshold* de 0.5. Então, a forma geral do modelo TS com uma abordagem MIMO (*multiple-input multiple-output*) [4], que explora uma combinação linear entre a entrada e os parâmetros, é apresentada da seguinte forma:

$$\hat{y}^i = f(\bar{x}, \Theta^i) = \bar{x}^T \Theta^i \quad (2)$$

$$\Theta^i = \arg \min_{\Theta^i} \sum_{i=1}^c \|y^i - X^i \Theta^i\|_2^2 \quad (3)$$

Com as variáveis de entrada \bar{x} sendo consideradas junto dos parâmetros dos consequentes das respectivas regras, os \hat{y}^i são normalizados, ou seja, $\sum_{i=1}^c \bar{y}^i = 1$, pois:

$$\bar{y}^i = \frac{\hat{y}^i}{\sum_{i=1}^c \hat{y}^i} \quad (4)$$

$$\bar{y} = \sum_{i=1}^c \frac{\mu^i}{\sum_{j=1}^c \mu^j} \bar{y}^i \quad (5)$$

Nessa situação, o rótulo final de cada classe $l, l = 1, \dots, K$ (sendo K o número de classes), segue o seguinte critério:

$$\hat{L}^l = \begin{cases} 1, & \text{se } \bar{y}_l \geq 0.5 \\ 0, & \text{caso contrário} \end{cases}$$

3. Classificação multi-rótulos com regressão e aprendizado multitarefa

Por último, como demonstrado por A. O. C. Ayres e F. J. Von Zuben [3], pode-se adicionar um termo de regularização ao cálculo dos parâmetros dos consequentes ($\Omega(\xi)$) que leva em consideração o aprendizado multitarefa, explorando a similaridade entre as regras através do grau de relação entre elas. Dessa forma a estrutura da regra TS segue a expressão a seguir:

$$\Theta^* = \arg \min_{\Theta} \sum_{i=1}^c \|y^i - X^i \Theta^i\|_2^2 + \Omega(\xi) \quad (6)$$

O aprendizado multitarefa, ao contrário de transferência de aprendizado (do inglês *transfer learning*), é utilizado, de modo que essas tarefas possam ser beneficiadas com o conhecimento que vai se formando entre elas, ao haver o compartilhamento dessa informação entre as tarefas, visando construir um aprendizado conjunto entre as regras.

3. Resultados e Discussões

Para encontrar o melhor conjunto de hiperparâmetros para cada versão, o *framework* Optuna foi utilizado, com a intenção de maximizar a função objetivo, buscando hiperparâmetros que entregassem o maior valor de *subset accuracy*. Para cada banco de dados, foram executados 100 *trials*. O intervalo dos hiperparâmetros para o conjunto de dados **emotions** foi: σ [0, 0.5], δ [1, 100], N [1, 50] e para a versão que utiliza o aprendizado multitarefa, ρ [0, 0.09]. Já o intervalo para o conjunto de dados **scene** foi: σ [0, 0.7], δ [1, 100], N [1, 100] e para a versão que utiliza o aprendizado multitarefa, ρ [0, 0.9].

Dois conjunto de dados foram utilizados para análise e ambos foram retirados do repositório **MULAN**. O **emotions** [7], é composto por 593 músicas descritas por 72 características. Cada conjunto de 593 instâncias pode pertencer a até 6 rótulos diferentes, baseados no modelo de Tellegen-Watson-Clark (*amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely e angry-fearful*). O outro conjunto de dados é o **scene** que possui cada uma das 2407 imagens descritas com 294 características numéricas correspondendo às cores espaciais das imagens convertidas no espaço CIELUV. Estas imagens possuem como rótulos possíveis *beach, sunset, fall foliage, field, mountain e urban*. Ademais, aplicou-se sobre os dados uma análise de componentes principais (PCA), um método não-supervisionado que reduz a dimensão dos dados de entrada a serem analisados, de modo a excluir *features* irrelevantes. Dessa forma, o conjunto de dados aplicados às três versões foram redimensionados, conforme especificado na **Tab. 1**.

Hyperparameter Importances

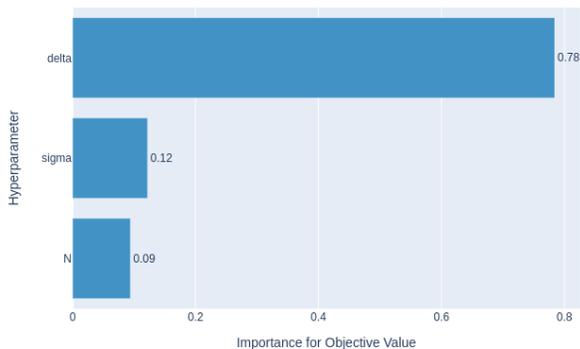


Fig. 1: Comparação da importância dos hiperparâmetros (v.1 - com regressão - emotions)

Hyperparameter Importances

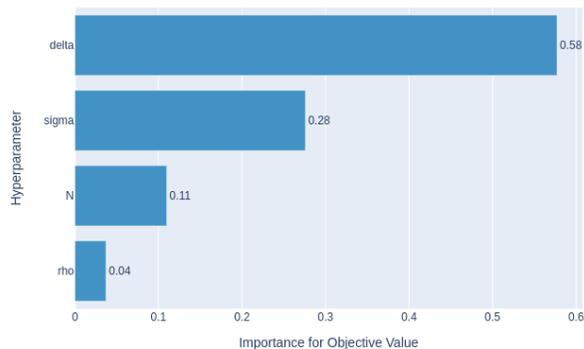


Fig. 2: Comparação da importância dos hiperparâmetros (v.2 - com regressão e aprendizado multitarefa - emotions)

Tab. 1: Descrição dos hiperparâmetros para os bancos de dados utilizados: *emotions*, *scene*

Dataset	dim	Rótulos	Hiperparâmetros (v.0)			Hiperparâmetros (v.1)			Hiperparâmetros (v.2)			
			σ	δ	N	σ	δ	N	σ	δ	N	ρ
Emotions [7]	10	6	0.470	47	5	0.476	93	48	0.472	53	15	0.026
Scene [8]	12	6	0.669	29	98	0.633	17	4	0.591	12	34	0.671

Nas Fig. 1 e Fig. 2 tem-se a comparação da importância dos hiperparâmetros para as versões v.0 e v.1, utilizando-se o banco de dados **emotions**. É possível notar que com a inserção do aprendizado multitarefa no cálculo dos parâmetros dos consequentes das regras, a importância do hiperparâmetro σ aumenta. Isso acontece porque a alteração no valor da granularidade do espaço *input-output* significa mais ou menos regras sendo formadas, o que impacta na possibilidade de aprendizado simultâneo entre as regras. Ademais, a importância significativa do hiperparâmetro δ , ou seja, do passo de atualização do modelo em ambas as versões, evidencia o impacto da adaptabilidade do algoritmo para uma boa performance.

Na Tab. 2 nota-se que o EVeC apresentou um dos melhores *subset accuracy* dentre os algoritmos [6] e o melhor valor de *precision* para ambos os bancos de dados utilizados. Ter um dos melhores *subset accuracy* dentre os algoritmos analisados indica que o EVeC é capaz de prever corretamente em uma boa parte de suas amostras todos os rótulos a que um conjunto de instâncias pertence, para o conjunto de dados analisados. Ademais, como a métrica *precision* apresenta o melhor valor, pode-se dizer, então, pela análise de ambas as métricas, que as previsões feitas pelo EVeC mostram-se consistentes.

Algoritmo	Emotions	Algoritmo	Emotions	Algoritmo	Scene	Algoritmo	Scene
EVeC v.0	0.508	EVeC v.0	0.114	EVeC v.0	0.824	EVeC v.0	0.745
EVeC v.1	0.612	EVeC v.1	0.252	EVeC v.1	0.936	EVeC v.1	0.855
EVeC v.2	0.643	EVeC v.2	0.257	EVeC v.2	0.958	EVeC v.2	0.883
HOMER	0.509	HOMER	0.163	HOMER	0.746	HOMER	0.661
ML-C4.5	0.606	ML-C4.5	0.277	ML-C4.5	0.592	ML-C4.5	0.533
RFML-C4.5	0.625	RFML-C4.5	0.272	RFML-C4.5	0.403	RFML-C4.5	0.372

(a) Precision

(b) Subset accuracy

(c) Precision

(d) Subset accuracy

Tab. 2: Comparação das métricas para os bancos de dados **emotions** e **scene**

Na Fig. 3, tem-se a variação no número de regras para as três versões com os hiperparâmetros assumindo os valores presentes na Tab. 1 e sendo aplicados sobre o conjunto de dados **emotions**. Nota-se uma redução no número de regras do modelo comparando-se os algoritmos. Ressalta-se que a versão com aprendizado multitarefa termina com um número de regras de 54, frente às 58 da versão que somente emprega a regressão de quadrados mínimos, o que indica que a utilização do aprendizado multitarefa se mostrou benéfica, uma vez que sua adição possibilitou melhores desempenhos em métricas *example-based* como *subset accuracy* e *precision*, mesmo que utilizando um menor número de regras.

Já na Fig. 4, para o conjunto de dados **scene**, a variação no número de regras para as três versões aplicadas ao conjunto de dados **scene** mostra-se muito mais dinâmica, o que permite visualizar a característica adaptativa durante todo o período com a criação, junção ou exclusão de regras ocorrendo conforme os dados chegam. Ao fim das versões v.1 e v.2, o número de regras se mostrou próximo, porém, conforme visto nas Tab. 2c e Tab. 2d, a versão que empregou o aprendizado compartilhado entre as regras aproveitou ao máximo o conjunto de dados, de modo a ter o melhor desempenho, tanto em termos de *precision* quanto de *subset accuracy*.

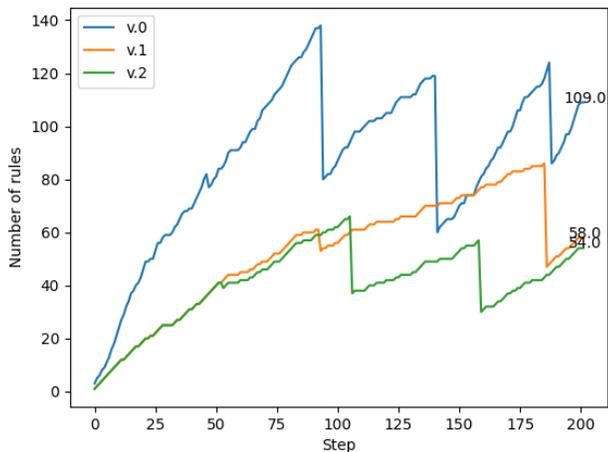


Fig. 3: Variação do número de regras durante o experimento para as versões EVeC (v.0), EVeC (v.1) e EVeC (v.2) para o banco de dados 'emotions' e hiperparâmetros indicados na Tab. 1

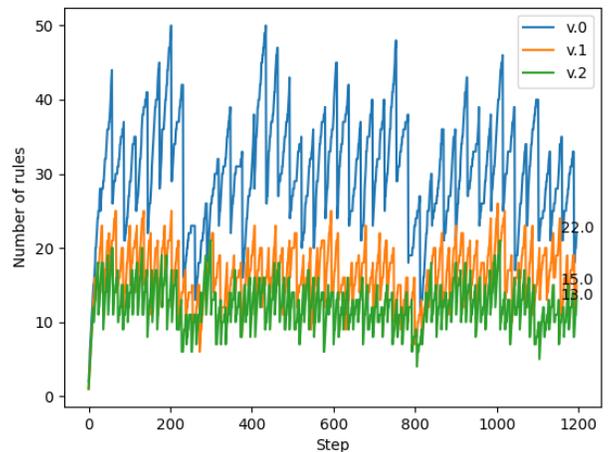


Fig. 4: Variação do número de regras durante o experimento para as versões EVeC (v.0), EVeC (v.1) e EVeC (v.2) para o banco de dados 'scene' e hiperparâmetros indicados na Tab. 1

4. Conclusão

O algoritmo EVeC mostra-se como uma ferramenta de classificação multi-rótulos em fluxo de dados com resultados competitivos frente a vários algoritmos da literatura, com a vantagem de ter os parâmetros e a estrutura adaptável, suprimindo a demanda crescente de classificadores para o tipo de dados em questão.

Ademais, ressalta-se que o uso do PCA para viabilizar a análise de um conjunto de dados de dimensão grande, poderia penalizar o EVeC. O que se observou, no entanto, foi que essa versão do algoritmo, mesmo utilizando um número de dimensões muito menor do que outros algoritmos da literatura, obteve resultados competitivos, o que evidência os benefícios de se utilizar a estratégia do aprendizado multitarefa.

Portanto, de modo geral, conclui-se que a adição de um termo de regularização na estimativa dos parâmetros dos consequentes das regras *fuzzy*, junto do erro empírico, assim, como mostrado por A. O. C Ayres e F. J. Von Zuben [2], [3], promove claros benefícios na aplicação de classificação multi-rótulos de fluxo de dados *online*.

5. Referências

- [1] Herrera, F., Charte, F., Rivera, A.J., del Jesus, M.J. Multilabel Classification: Problem Analysis, Metrics and Techniques, Springer, 2016.
- [2] Ayres, A.; Von Zuben, F. The Extreme Value Evolving Predictor. *IEEE Transactions on Fuzzy Systems*, 2020
- [3] Ayres, A.; Von Zuben, F. Multitask learning applied to evolving fuzzy-rule-based predictors. *Evolving Systems*, vol. 12, pp. 407–422, 2021.
- [4] Angelov P. Plamen; Evolving Fuzzy-Rule-Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1462–1475, 2008
- [5] Yedla A. A Tutorial on Recursive methods in Linear Least Squares Problem, Duke University, 2008
- [6] Zheng, X.; Peipei, L.; Chu, Z.; Hu, X. A Survey on Multi-Label Data Stream Classification. *IEEE Access*. vol. 8, pp. 1249–1275, 2020.
- [7] Wiczorkowska, A.; Synak, P.; Ras, Z.W.. Multi-Label Classification of Emotions in Music. In *Intelligent Information Processing and Web Mining*, pp. 307–315, Springer, 2006.
- [8] Boutell, M.; Luo, J.; Shen, X.; Brown, C. Learning multi-label scene classification. In *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.