Não existe almoço grátis!

Redes Neurais Profundas

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR

Professor

Paulo Ricardo Lisboa de Almeida
Doutor em Ciência da Computação, Bel. em Eng. da Computação.

paulorla@ufpr.br

prlalmeida.com.br

www.linkedin.com/in/paulorla

Professor - Pesquisa

Aprendizado de máquina.

Machine Learning para fluxos de dados.

Cidades inteligentes.

DSBD

Nosso Hardware

Paulo

Grégio

Zanata

DSBD + Secret + Hipes

4 servidores:

- 328 processadores.
- 3,3 TB de DRAM.
- GPUs para criação de modelos de IA
  - 4 GPUs NVidia A5000.
  - 1 GPU NVidia A6000
  - 43.520 CUDA Cores.
  - 144 GB de memória de vídeo.

Efficient Prequential AUC-PR Computation

Artigo - IEEE ICMLA 2023.

Florida, Estados Unidos.

David

Paulo

Grégio

Zanata

Efficient Prequential AUC-PR Computation

David L. Pereira Gomes, André Grégio, Marco A. Zanata Alves, Paulo R. Lisboa de Almeida
Department of Informatics – Federal University of Parana (UFPR) – Curitiba, PR - Brazil
david.gomes@ufpr.br, gregio@ufpr.br, mazalves@ufpr.br, paulorla@ufpr.br

***Abstract*—When dealing with classification problems for data streams, we often need to compute the classification metrics in a prequential manner. The Area Under the Precision-Recall Curve (AUC-PR) metric is extensively used in imbalanced classification scenarios, where the negative class outnumbers the positive one. Despite its advantages, it may be computationally expensive to recompute that metric every time a new test instance becomes available. In this work, we present an efficient algorithm to compute the AUC-PR in a prequential way. Our proposed algorithm uses a self-balancing binary search tree to avoid the need to reorder the data when updating the AUC-PR value with the most recent data. Our experiments take into consideration six well-known, publicly available stream-based datasets. Our experiments show that our approach can be up to 13 times faster and use 12 times less energy than the traditional batch approach when considering a window of size 1,000.**

***Index Terms*—AUC-PR, prequential, stream, metrics**

I. Introduction

The massive amount of data produced by sensors, devices, and users poses a challenge to the application of classification algorithms whose output needs to be provided in real-time (e.g., critical systems, emergency diagnosis, security, threat detection, etc.). Those data are usually temporally-dependent, arriving at the classifier as a data stream.

Metrics for classification problems involving data streams, such as accuracy, F1-score, and Area Under the Precision-Recall Curve (AUC-PR), are often computed in a prequential manner, i.e., every time a new test instance becomes available. The reasoning behind the prequential calculation of those metrics is to allow for the monitoring of the classifier's performance over time, as well as quickly reacting to environmental changes (e.g., concept drifts), which may hinder the classification capability of a decision-support system.

Therefore, the prequential computation of classification metrics often requires computing them using a window W that contains the latest labeled data received. Thus, a metric must be recomputed on each update of this window, which may lead to an overhead that turns the classification of data streams into an overly expensive task, especially if we rely on computationally intensive metrics, such as the AUC-PR. The incurred overhead may increase costs (e.g., more computing power in servers) and the carbon footprint associated with this type of classification system.

In this work, we introduce an efficient algorithm to compute the AUC-PR for streams in a prequential manner (assuming a stream of instances, in which samples arrive for classification one at a time). The AUC-PR metric belongs to a family of metrics focused on imbalanced scenarios. To the best of our knowledge, this is the first algorithm to reduce the time complexity from $O(m \log m)$ to $O(m)$ when computing the AUC-PR metric for streams. In our experiments, the proposed algorithm was 13 times faster and used 12 times less energy when compared to the batch approach (i.e., recomputing the metric from scratch every time the window W is updated), often used when a prequential algorithm is unavailable. The main contributions of this paper are:

- An algorithm to calculate the AUC-PR for stream settings in a prequential way, focusing on its efficiency;
- The evaluation of our proposed algorithm and comparison with a widely used implementation of the metric that considers batch settings.

II. Background and Related Work

In this Section, we introduce concepts needed for properly understanding our proposed method, such as the prequential computation of metrics and the definition of the Precision-Recall Curve. We also present the related state-of-the-art work.

A. Batch versus Prequential Metrics

Data classification can be divided into two settings: batch (or static) learning or stream. In the former, we consider that the available data is limited to a "snapshot" of a certain period of time, whereas in the latter, we have to handle unlimited data continuously arriving at potentially high rates [1].

Under a static setting, we may create a classifier using a train set S_c, and test its performance using some metric in a test set S_t, where $S_c \cap S_t = \emptyset$. This approach is known as holdout or batch testing [2]. On the other hand, under a stream scenario, new instances arrive over time, making it impossible to have a fixed test set to assess the classifier's performance—especially under conditions where the problem may evolve.

In a stream scenario, it is common to define a window W containing the m latest instances received and compute the performance metrics using this window. Every time a new test instance arrives, this window is moved to accommodate the new instance, and the metrics are updated. This approach is known as the prequential computation of the metric [2].

For example, let's consider the stream at times t and $t+1$ in Figure 1, in which the metrics are computed within a window that contains the $m = 5$ latest instances. When a test instance $\mathbf{x}_{t+1}$ arrives at time $t+1$, the window is moved to accommodate this new instance, and the oldest instance in the window ($\mathbf{x}_{t-4}$) is removed from the window.

When the window moves, it is updated, and we may recompute the performance metrics using the entire window (i.e., the whole window is considered a batch). Besides being simple,

ICMLA 2023.

Vehicle Occurrence-based Parking Space Detection

Artigo - IEEE SMC 2023.

Havaí, Estados Unidos.

Vehicle Occurrence-based Parking Space Detection

Paulo R. Lisboa de Almeida[1], Jeovane Honório Alves[1], Luiz S. Oliveira[1], Andre Gustavo Hochuli[2], João V. Fröhlich[3], and Rodrigo A. Krauel [3]

***Abstract*— Smart-parking solutions use sensors, cameras, and data analysis to improve parking efficiency and reduce traffic congestion. Computer vision-based methods have been used extensively in recent years to tackle the problem of parking lot management, but most of the works assume that the parking spots are manually labeled, impacting the cost and feasibility of deployment. To fill this gap, this work presents an automatic parking space detection method, which receives a sequence of images of a parking lot and returns a list of coordinates identifying the detected parking spaces. The proposed method employs instance segmentation to identify cars and, using vehicle occurrence, generate a heat map of parking spaces. The results using twelve different subsets from the PKLot and CNRPark-EXT parking lot datasets show that the method achieved an AP25 score up to 95.60% and AP50 score up to 79.90%.**

I. INTRODUCTION

Smart-parking solutions use sensors, cameras, and data analytics to optimize parking spaces. The goal is to reduce traffic congestion and make finding a parking spot more efficient. Displaying real-time information about the availability of parking spots to drivers, either through a smartphone app or other means, is a key feature of smart parking solutions.

In the last decade, extensive research has been conducted utilizing computer vision-based methods to tackle the problem of parking lot management. In this context, public datasets have been proposed [1]–[3] and several different approaches ranging from shallow to deep learning have been reported to detect and classify parking spaces. Systematic reviews can be found in [4] and [5].

As pointed out in [5], most of the works published in the literature assume that the parking spots are manually labeled, and all the system needs to do is to indicate if there is a car in that parking spot. However, the difficulty and time required for labeling images can impact the cost and feasibility of deploying computer vision systems.

In this scenario, automatic parking space detection is mandatory to facilitate the deployment of smart parking solutions. This task is depicted in Figure 1, where the yellow coordinates of the rotated rectangles (defining each spot) should be automatically defined. It is a challenging task since parking spaces are similar to roads, i.e., how can a model discriminate between a parking space and a road segment? The presence of cars may hinder correct detection, especially for methods that rely on the painted demarcations in the parking lots that delimit the parking spaces.

Fig. 1: Parking space locations (image from CNRPark-EXT).

The literature shows some efforts towards automatic parking space detection, which can be classified into classical image processing methods [6], [7], grid-based methods [3], [8], and deep learning [9]–[11]. As pointed out in [5], most works did not address the problem directly but discussed it as an intermediary step for the classification system. Consequently, they do not provide quantitative results, making it impossible to compare different works.

The main contribution of this work is a method specific to automatic parking space detection, which receives as input a sequence of images of a parking lot and returns a list of coordinates of rotated rectangles identifying the detected parking spaces. The rationale is based on the premise that parking spaces are regions where vehicles remain stationary for extended periods. The proposed method employs instance segmentation to identify cars, which is then utilized to generate an automatic heat map of parking spaces.

Through comprehensive experimental results on twelve parking lot subsets from the PKLot and CNRPark-EXT datasets, we demonstrate that the proposed method can detect parking spaces without prior knowledge, avoiding the laborious task of manual segmentation. The results show that the method achieved an average precision at an Intersection over Union (IoU) threshold of 25% (AP25) up to 95.60%, and an average precision at an IoU threshold of 50% (AP50) up to 79.90%. As discussed in our experiments, these metrics can be impacted by illegal parking and unused parking spaces. Although this affects the AP metric, it highlights the method's adaptability to dynamic parking scenarios, such as seasonal events, without prior knowledge of demarcations. Besides, our results compare favorably to the literature.

This work has been supported by the Brazilian National Council for Scientific and Technological Development (CNPq) – Grant 405511/2022-1

[1]Department of Informatics, Federal University of Paraná (UFPR), Curitiba, Brazil paulorla@ufpr.br, jeohalves@gmail.com, luiz.oliveira@ufpr.br

[2]Graduate Program in Informatics, Pontifical Catholic University of Paraná (PUCPR), Curitiba, Brazil. aghochuli@ppgia.pucpr.br

[3]Department of Computer Science, Santa Catarina State University (UDESC), Joinville, Brazil joaovitorfrohlich@gmail.com, rodrigoakrauel@gmail.com

SMC 2023.

UFPR

Paulo

Luiz

PUC PR

Jeovane

André

UDESC

João

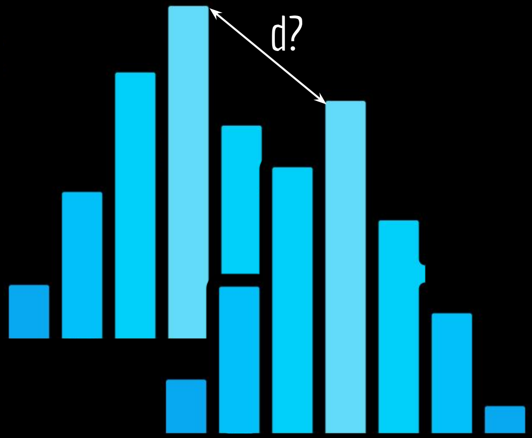
Rodrigo

Statistical Validation of Column Matching in the Database Schema Evolution of the Brazilian Public School Census

Parceria com o MEC via C3SL.

Como compatibilizar dados de diferentes anos do censo escolar usando métodos de *goodness of fit*?

SBBD 2024.



Muriki

Diogo

Simone

Paulo

Eduardo

Professor - Pesquisa

Grupo DSBD.

- **dsbd.inf.ufpr.br**

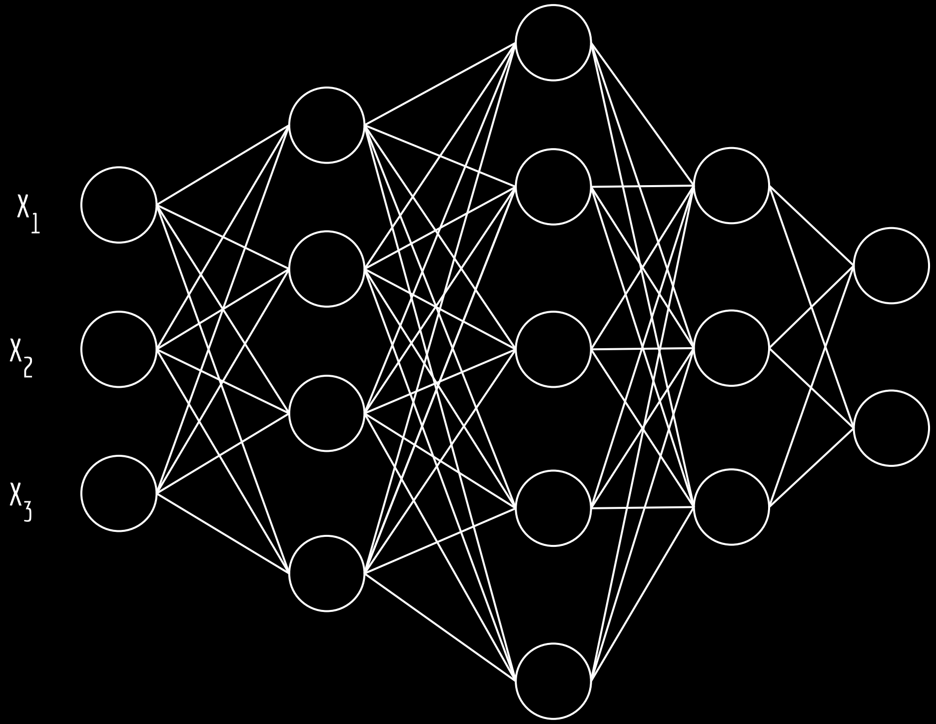
Tratamos mais sobre o assunto no final da disciplina.

- Iniciação científica, TCC, Mestrado, Doutorado, ...

Deep Learning

Redes neurais com muitas camadas são denominadas profundas – *Deep Neural Networks.*

A área do Aprendizado de Máquina que lida com essas redes é chamada de *Deep Learning* (Aprendizado Profundo).



Deep Learning

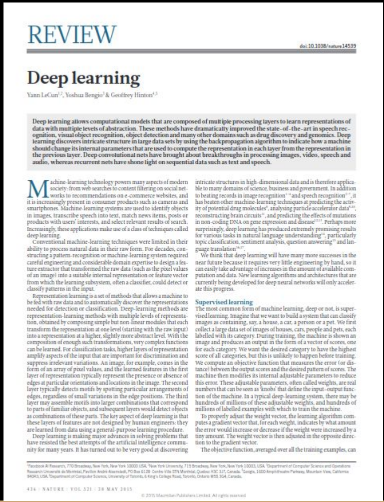
Métodos de aprendizado tradicional possuem limitações com dados crus.

- Precisam de especialistas de domínio para transformar o dado cru em algo que pode ser processado.
  - Extratores de Características.

Métodos de Deep Learning podem "aprender automaticamente" a representação dos dados.

- Gerar os extratores de características internamente.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553: 436-444. 2015.


REVIEW

doi:10.1038/nature14539

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning.

Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a pattern-recognition or machine-learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input.

Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned. For classification tasks, higher layers of representation amplify aspects of the input that are important for discrimination and suppress irrelevant variations. An image, for example, comes in the form of an array of pixel values, and the learned features in the first layer of representation typically represent the presence or absence of edges at particular orientations and locations in the image. The second layer typically detects motifs by spotting particular arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure.

Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition[1–4] and speech recognition[5–7], it has beaten other machine-learning techniques at predicting the activity of potential drug molecules[8], analysing particle accelerator data[9,10], reconstructing brain circuits[11], and predicting the effects of mutations in non-coding DNA on gene expression and disease[12,13]. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding[14], particularly topic classification, sentiment analysis, question answering[15] and language translation[16,17].

We think that deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are currently being developed for deep neural networks will only accelerate this progress.

Supervised learning

The most common form of machine learning, deep or not, is supervised learning. Imagine that we want to build a system that can classify images as containing, say, a house, a car, a person or a pet. We first collect a large data set of images of houses, cars, people and pets, each labelled with its category. During training, the machine is shown an image and produces an output in the form of a vector of scores, one for each category. We want the desired category to have the highest score of all categories, but this is unlikely to happen before training. We compute an objective function that measures the error (or distance) between the output scores and the desired pattern of scores. The machine then modifies its internal adjustable parameters to reduce this error. These adjustable parameters, often called weights, are real numbers that can be seen as 'knobs' that define the input–output function of the machine. In a typical deep-learning system, there may be hundreds of millions of these adjustable weights, and hundreds of millions of labelled examples with which to train the machine.

To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the error would increase or decrease if the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector.

The objective function, averaged over all the training examples, can

[1]Facebook AI Research, 770 Broadway, New York, New York 10003 USA. [2]New York University, 715 Broadway, New York, New York 10003, USA. [3]Department of Computer Science and Operations Research Université de Montréal, Pavillon André-Aisenstadt, PO Box 6128 Centre-Ville STN Montréal, Quebec H3C 3J7, Canada. [4]Google, 1600 Amphitheatre Parkway, Mountain View, California 94043, USA. [5]Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 3G4, Canada.

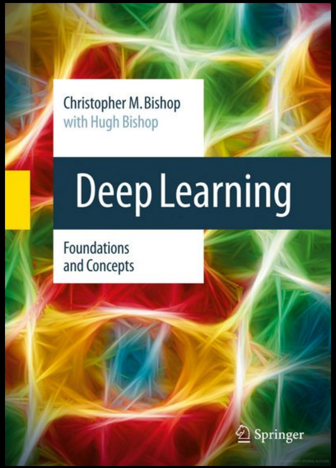
Deep Learning

Algumas redes do estado da arte:

Rede	Parâmetros	Memória (float 32 bits)
MobileNetV3	5 Milhões	0,02 GB
ResNet152	6 Milhões	0,22 GB
vit_l_32	306 Milhões	1,14 GB
Llama 3 8B	8 Bilhões	27 GB
Llama 3 70B	70 Bilhões	261 GB
GPT-3	175 Bilhões	651 GB

Não existe almoço grátis

"... pode parecer [que as redes neurais profundas] representam um algoritmo de aprendizagem universal, capaz de resolver qualquer tarefa ... [mas] o teorema que diz que não existe almoço grátis, diz que qualquer algoritmo de aprendizagem de máquina é tão bom quanto qualquer outro, fazendo-se a média dos problemas possíveis. Se um modelo é melhor do que outro para alguns problemas, ele certamente é pior que os demais em outros problemas ..."



Leia a Seção 9.1.2 No free lunch theorem em Bishop, C. M., Bishop, H. Deep Learning: Foundations and Concepts. 2023.

Ementa

- Álgebra e Cálculo aplicado a Redes Neurais.
- Perceptrons e Redes Neurais rasas.
- Técnicas de Treinamento.
- Redes Neurais Multicamadas.
- Redes Neurais Profundas.
- Redes Convolucionais.
- Transformers.
- Normalização e tratamento de dados.
- Funções de custo.

UFPRVirtual

A disciplina tem uma página oficial na UFPRVirtual.

Será o meio oficial de comunicação.

Atualize o seu e-mail.

Para se comunicar, você pode ir até a sala do professor, ou enviar e-mail.

paulorla@ufpr.br

Conhecimentos Necessários

Se você já cumpriu a barreira, tem conhecimentos o suficiente para a disciplina.

- Álgebra Linear.
- Cálculo Diferencial e Integral.
- Algoritmos, programação e estruturas de dados.
- Peculiaridades do hardware e ajustes para desempenho (organização da memória, barramentos, GPUs, ...).

Avaliação

Prova 1: 30%

Prova 2: 30%

Trabalho: 40%

Mínimo de presenças: 75% -> **reprovado** automaticamente se não cumprir.

Aprovado se média ≥ 70.

Exame se nota ≥ 40. Nesse caso nota = (média + exame)/2. **Aprovado** se nota ≥ 5.

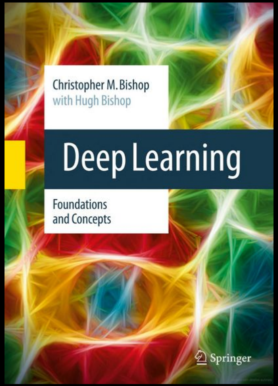
Entregas e Provas

Não serão aceitas entregas em atraso (exceto casos amparados pela UFPR).

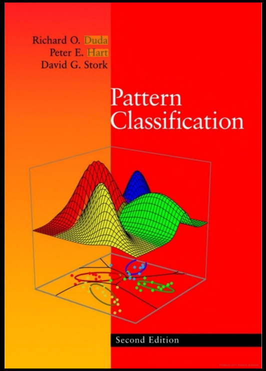
Em caso de plágio, **todos envolvidos** ficam com zero.

Bibliografia Básica

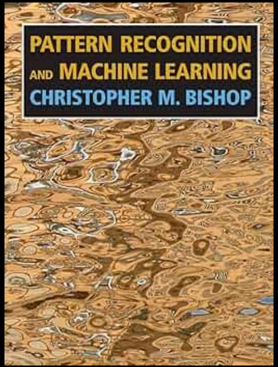
Bishop, C. M., Bishop, H. Deep Learning: Foundations and Concepts. 2023.



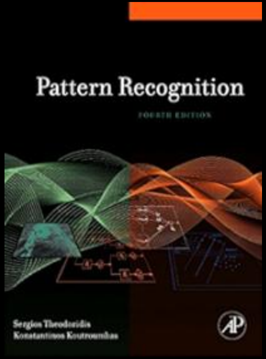
Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification. 2012.



Bishop, C. M. Pattern Recognition and Machine Learning. 2006.



Theodoridis, S., Koutroumbas, K. Pattern Recognition & Matlab Intro. 2010.



Biblioteca Virtual

A UFPR dá acesso gratuito a uma biblioteca virtual a todos seus alunos.

Acesse via https://minhabiblioteca.ufpr.br/biblioteca

Dica: você vai precisar do seu e-mail @ufpr para acessar.

Ficha 2

Para mais detalhes, como bibliografia complementar e programa completo da disciplina, veja a Ficha 2.

É um plano, portanto pode sofrer alterações.

Em especial, as datas podem mudar.

Tudo será avisado com antecedência.

Licença