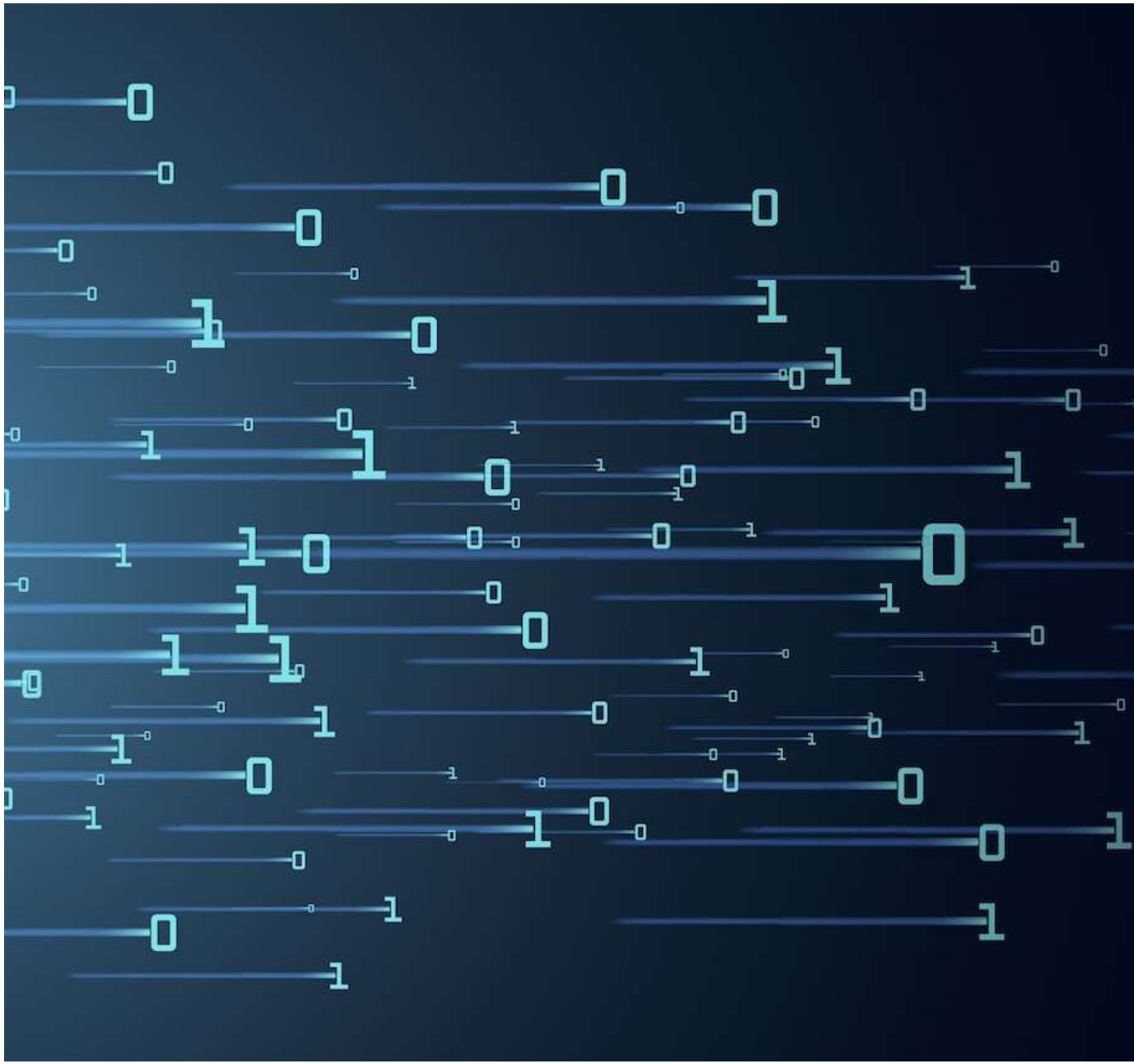
PyCCD

Análise do desempenho e dos tempos de computação

Reunião 23/05/2024

Sara Caetano, Daniel Moraes e Manuel Campagnolo



Índice

Parâmetros do CCD

- **'PEEK_SIZE':** 6
- **'MIN_YEARS' :** 1
- **'DETECTION_BANDS':** NDVI, Green, SWIR2
- **'TMASK_BANDS':** Green, SWIR2
- **'LASSO_MAX_ITER':** 25000
- **'ALPHA':** 200
- **'CHISQUAREPROB':** 0.999

Moraes, D., Barbosa, B., Costa, H., Moreira, F. D., Benevides, P., Caetano, M., & Campagnolo, M. (2024). Continuous forest loss monitoring in a dynamic landscape of Central Portugal with Sentinel-2 data. *International Journal of Applied Earth Observation and Geoinformation*, *130*, 103913. https://doi.org/10.1016/j.jag.2024.103913

ÚLTIMA REUNIÃO (10 000 pixels):

	F1-Score [%]	Omission Error [%]	Commission error [%]
PyCCD			
THEIA/MAJA	78.57	22.20	20.64
GEE/s2cloudless	71.48	33.97	22.09
eeCCD			
GEE/s2cloudless	81.14	15.77	21.73

Desempenho do CCD sobre dados de referência

Para melhorar o desempenho do modelo, foi feito uma série de testes para determinar o valor ideal do parâmetro alpha, que desempenha um papel fundamental na regressão Lasso usada no PyCCD.

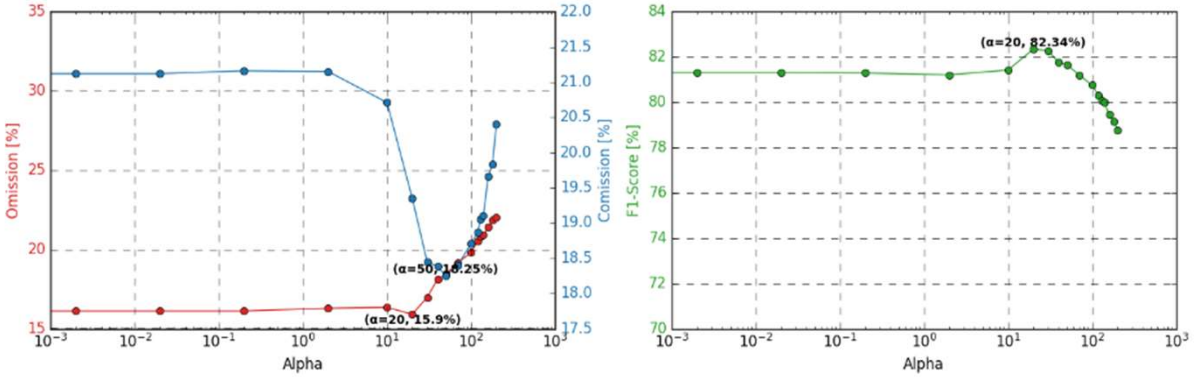
DESEMPENHO ATUAL DO CCD (10 000 pixels):

	F1-Score [%]	Omission Error [%]	Commission error [%]
PyCCD			
THEIA/MAJA	82.34	15.90	18.38
GEE/s2cloudless	83.36	15.32	17.92
eeCCD			
GEE/s2cloudless			21.73

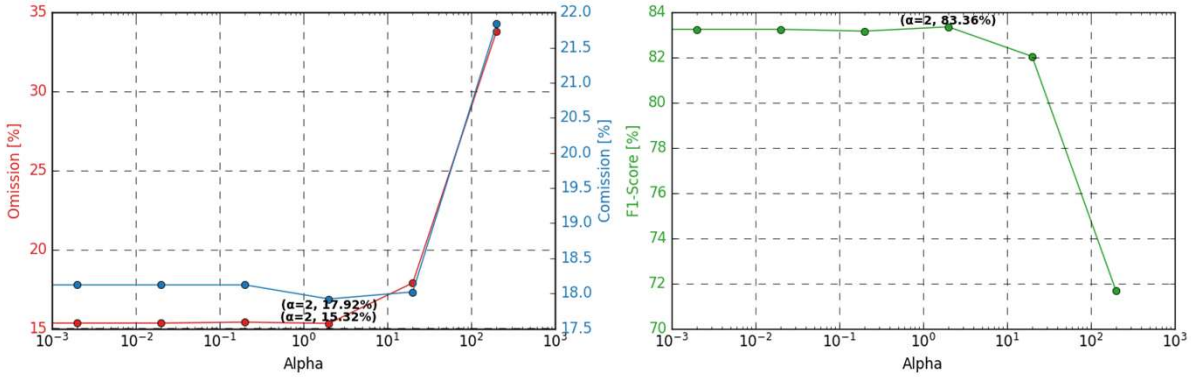
PyCCD (Theia): alpha = 20

PyCCD (GEE): alpha = 2

THEIA/MAJA

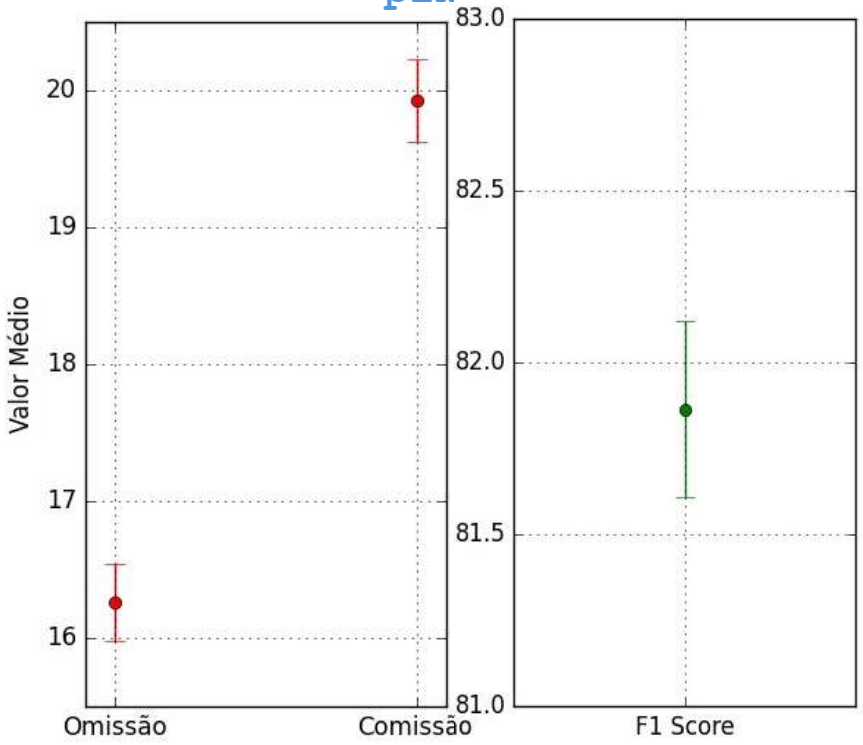


GEE/S2CLOUDLESS



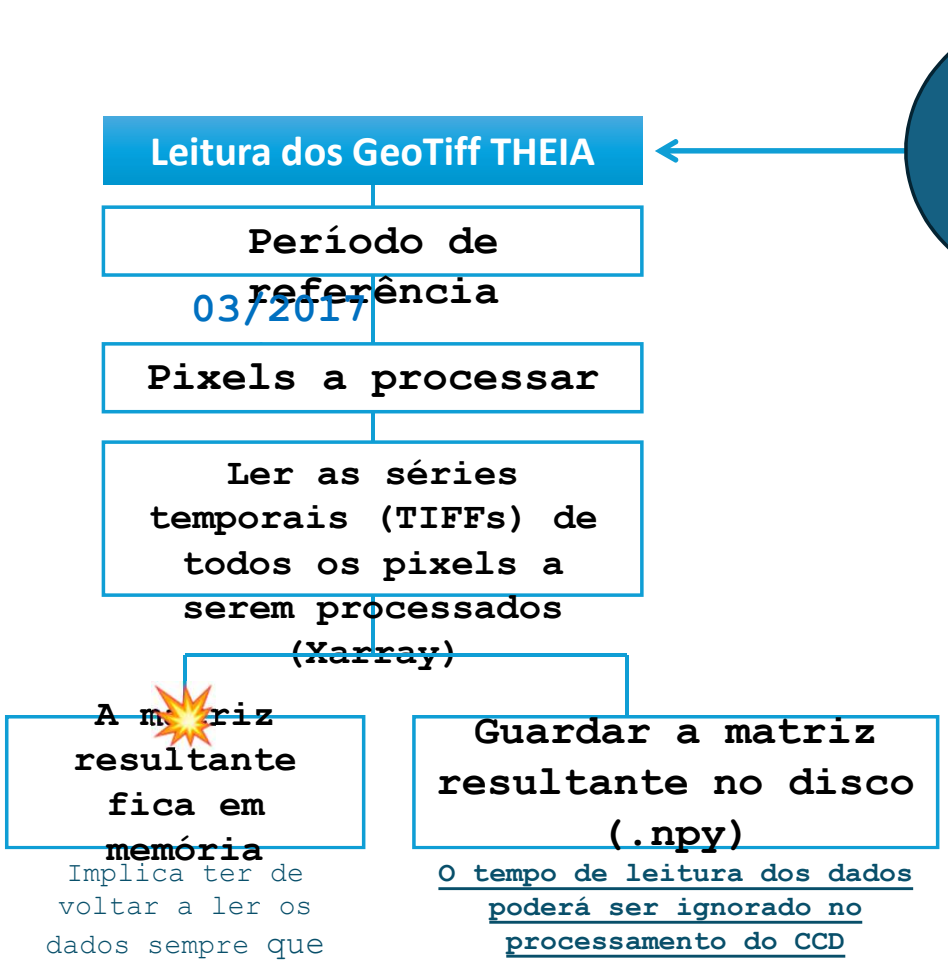
Desempenho do CCD sobre dados de referência

30 "RUNS" DE 10 000 PIXELS (BDR = 241 941 pixels)



	Intervalo de confiança	Valor médio
Omissão	(15.99, 16.54)	16.26
Comissão	(19.62, 20.23)	19.93
F1 Score	(81.61, 82.12)	81.86

Aspetos computacionais do PyCCD



Processamento do CCD

PROCESSOS RESPONSÁVEIS PELA MAIOR PARCELA DO TEMPO DE COMPUTAÇÃO:

- **Regressão Robust Iteratively Reweighted Least Squares:** TMask **~ 20%** do tempo de processamento
- **Regressão LASSO: ~ 77%** do tempo de processamento do CCD (função principal do algoritmo)

Aspetos computacionais: tempo de leitura dos ficheiros GeoTiff

Testes realizados numa máquina equipada com processador **i7-8700 3.20GHz 6 Core(s) e 64 GB de RAM**, em que o **PERÍODO DE PROCESSAMENTO DO PyCCD** foi de 03/2017 a 12/2021.

Número de pixels	Tempos de leitura dos Tiffs com Xarray
10 000	~ 10 mins
100 000	~ 38 mins
240 000	~ 116 mins

\+ pixels

\- tempo de execução

Aspetos computacionais: tempo de computação CCD

Testes realizados numa máquina equipada com processador **i7-8700 3.20GHz 6 Core(s) e 64 GB de RAM**, em que o **PERÍODO DE PROCESSAMENTO DO PyCCD** foi de **03/2017** a **12/2021.**

	'ALPHA'	'LASSO_MAX_ITER'	Tempo de processamento CCD (10 000 pixels)	Desempenho do CCD	
Regressão linear	0	1000	15.39 mins	F1-score = 80.24% Omission error = 16.59% Commission error = 22.69%	- tempo - desempenho
Função LASSO	20	1000	17.55 mins	F1-score = 81.82% Omission error = 16.22% Commission error = 20.05%	+ tempo + desempenho
	20	25000	18.18 mins	F1-score = 81.82% Omission error = 16.22% Commission error = 20.05%	+ tempo = desempenho

Aspetos computacionais: Estimativa do tempo total para Portugal Continental

Testes realizados numa máquina equipada com processador **i7-8700 3.20GHz 6 Core(s) e 64 GB de RAM**

Área que é simultaneamente Floresta ou Mato na COS 2018 e na COSc 2023 (intersecção) = 352 milhões e 116 mil pixels

Máscara

EXCLUÍ-SE: Pixels com **NDVI>0.7** e pixels em que o **NDVI subiu ou é igual ao mês anterior** (esta regra foi aplicada aos compósitos mensais da DGT).

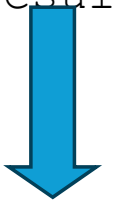
	Número de pixels	Tempo de computação do CCD (*)
Outubro 2023	56.409.000	58 dias
Novembro 2023	45.207.000	47 dias
Dezembro 2023	54.558.000	56 dias
Janeiro 2024	81.819.000	85 dias

(*) Os tempos estimados foram calculados usando o valor de referência de 15.39 mins para 10000 pixels (excluindo a leitura dos dados).

Estratégias a explorar para reduzir o tempo de computação

ASPETOS A CONSIDERAR QUE AUMENTAM O TEMPO DO CCD

- **Armazenamento dos dados dos pixels na memória:** embora a leitura se torne mais rápida ao ler um maior número de pixels, durante o processamento do CCD, a memória pode ficar sobrecarregada com os dados, resultando num processamento mais lento.



SOLUÇÕES A EXPLORAR:

1. Guardar os dados lidos num ficheiro temporário no disco e processar o CCD em "*batchs*" de 10 000 pixels para acelerar o processo.

Número de pixels	Tempo de execução com os dados em memória (*)	Tempo de execução com os dados guardados no disco (.npy) (*)	Tempo de execução apenas para o CCD usando dados guardados no disco (.npy)
100 000	16 horas e 30 mins	5 horas e 44 mins	5 horas e 6 mins

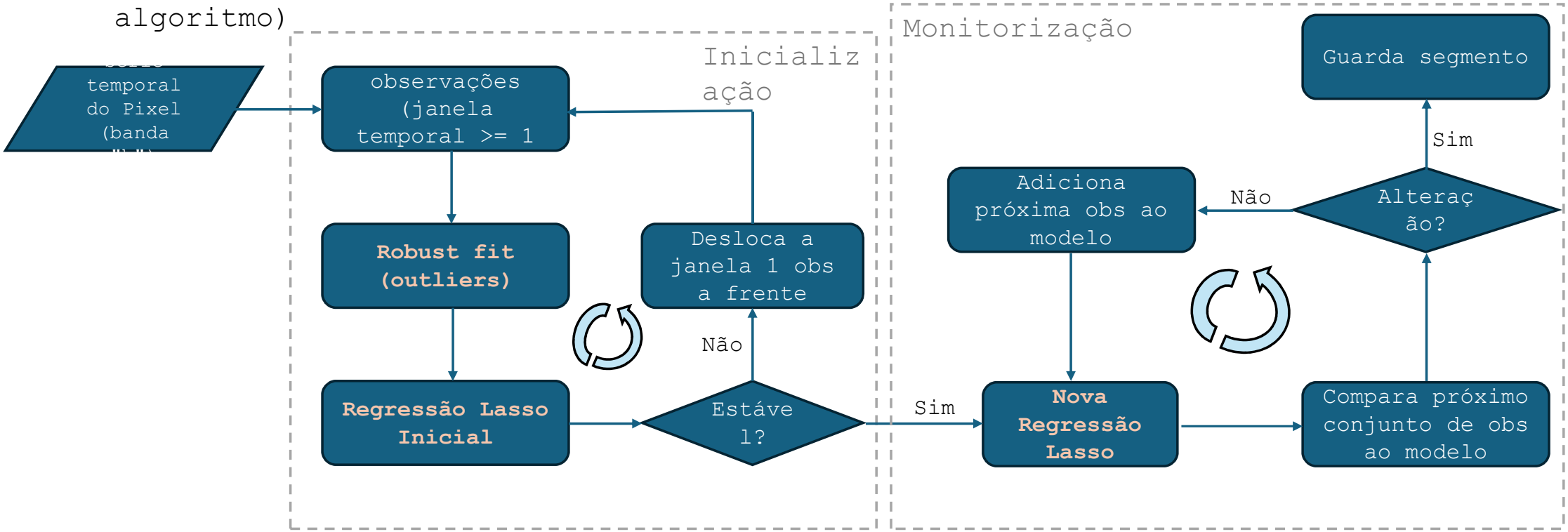
(*) estes tempos incluem a leitura dos tiffs + processamento do CCD c/ alfa = 20

Estratégias a explorar para reduzir o tempo de computação

PROCESSOS RESPONSÁVEIS PELA MAIOR PARCELA DO TEMPO DE COMPUTAÇÃO:

- **Regressão Robust Iteratively Reweighted Least Squares:** TMask ~20% do tempo de processamento

- **Regressão LASSO:** ~77% do tempo de processamento do CCD (função principal do algoritmo)



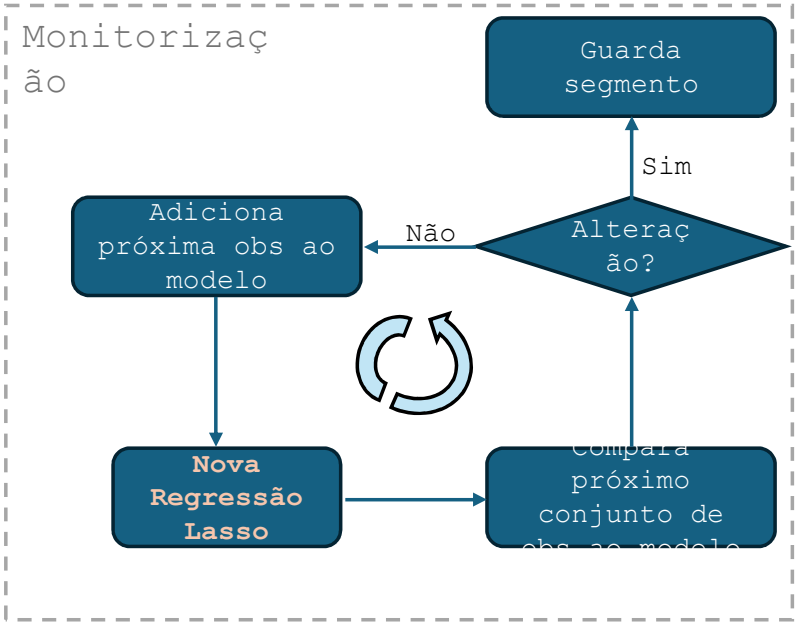
Estratégias a explorar para reduzir o tempo de computação

I. ADAPTAÇÃO DO ALGORITMO PyCCD:

- Reduzir chamadas à função LASSO
- Adicionar grupos de observações ao invés de observação individual

II. IDENTIFICAÇÃO DE PIXELS COM TRAJETÓRIA SEMELHANTE UTILIZANDO MACHINE LEARNING

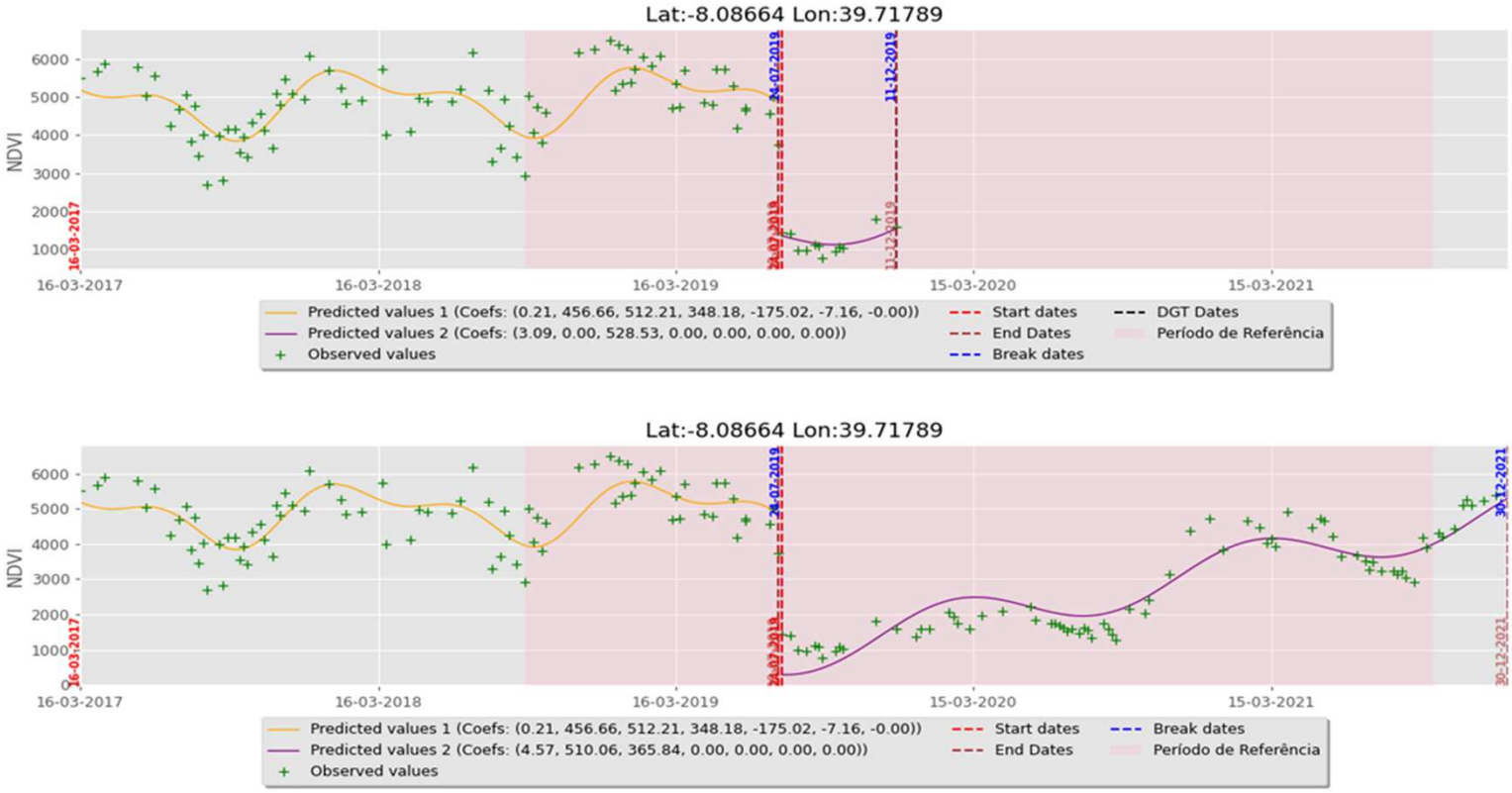
- "Protótipo" de estratégia - requer testes
- Treinar um classificador para agrupar pixels com trajetórias semelhantes
- Correr PyCCD para apenas um pixel representativo do grupo
- Transmitir informação (data da quebra, magnitude *etc*) do pixel representativo para o grupo



Estratégias a explorar para reduzir o tempo de computação

III. REAPROVEITAR OS SEGMENTOS JÁ CALCULADOS

- Correr o PyCCD apenas a partir da última data de quebra
- Evita correr o algoritmo desde o princípio sempre que se pretenda estender a série temporal



Potenciais adaptações tecnológicas a implementar na cadeia de produção da DGT

- Pré-processamento da leitura dos dados

- Os testes realizados devem considerar as capacidades das máquinas, como o número de cores e a velocidade do processador.

- Criar versão melhorada do algoritmo de forma a correr de forma eficiente nas máquinas da DGT.