Estruturas de Dados Probabilísticas

O que são e como podem ajudar nos desafios de arquitetura de dados e privacidade?

inloco

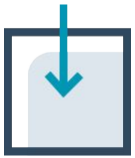
Lucas Alves Rufino

Bacharel em Ciência da Computação pela Universidade Federal de Pernambuco.

Engenheiro de dados na In Loco.

Pesquisador na área de mineração de opinião e aprendizagem de máquina.

inloco



Estamos em mais 60 milhões de smartphones.

Mais de 16 TB de dados processados diariamente.

Temos mais de 28 milhões de lugares mapeados no mundo.

Nosso propósito

Entregar conveniência para as pessoas garantindo sua privacidade.

Roteiro

Motivação
Desafios de arquitetura de dados e privacidade.

Estruturas de dados probabilísticas
Pertencimento, frequência e contagem distinta.

Aplicações
Exemplos, resultados e melhorias.

Motivação

1.
Atualidade

Estamos na era orientada a dados.

2.
Warehouse

Dado acessível e consolidado para uso.

3.
Performance

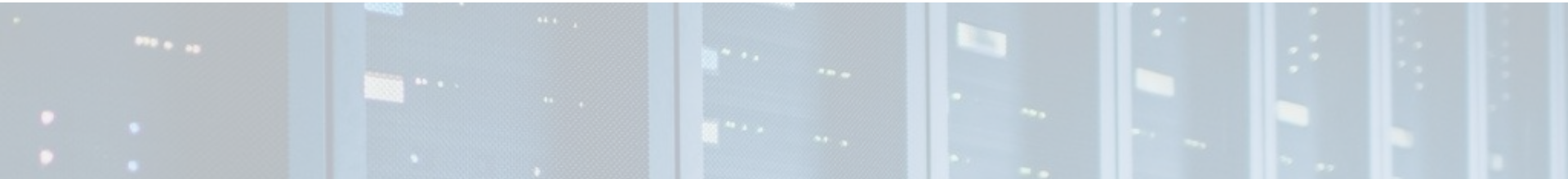
Guardar e processar é custoso.

4.
Anonimato

Garantindo a privacidade do usuário.

5.
Retenção

Salvando dados para uso a longo prazo.



Estruturas de dados probabilísticas

Atualidade
Uma estratégia orientada a dados para a era dos dados.

Warehouse
Estruturas consolidadas geram informações consolidadas.

Performance
Barato para guardar e ainda mais para processar.

Anonimato
Permite aplicar estratégias de anonimização facilmente.

Retenção
Se a estrutura está anonimizada, então é seguro guardar.

Conjuntos representando problemas

Pertencimento

Dado um elemento *x* e um conjunto *C*, como testar se *x* pertence a *C*?

$|\{x\} \cap C| == 1$

Frequência

Dado um elemento *x* e *n* conjuntos C_i como contar quantas vezes x aparece nos conjuntos?

$sum(i=0, n)|\{x\} \cap C_i|$

Cardinalidade

Dado um conjunto *C*, como calcular o seu tamanho?

$|C|$

Problema do pertencimento

Solução imediata:

Criar uma estrutura de dados (*Set* / *HashSet*) com todos os elementos.

Proposta alternativa:

- Amanda Melo Araujo
- Carolina Correia Castro
- Felipe Martins Castro
- Nicolas Gomes Ferreira

- Carolina Correia Castro
- Igor Melo Cardoso
- Felipe Correia Araujo

Problema do pertencimento

Solução imediata:

Criar uma estrutura de dados (*Set* / *HashSet*) com todos os elementos.

Proposta alternativa:

- Amanda Melo Araujo
- Carolina Correia Castro
- Felipe Martins Castro
- Nicolas Gomes Ferreira

- Carolina Correia Castro
- Igor Melo Cardoso
- Felipe Correia Araujo

- Amanda
- Araujo
- Carolina
- Castro
- Correia
- Felipe
- Martins
- Melo

Problema do pertencimento

Solução imediata:

Criar uma estrutura de dados (*Set* / *HashSet*) com todos os elementos.

Proposta alternativa:

- Amanda Melo Araujo
- Carolina Correia Castro
- Felipe Martins Castro
- Nicolas Gomes Ferreira

- Carolina Correia Castro
- Igor Melo Cardoso
- Felipe Correia Araujo

- Amanda
- Araujo
- Carolina
- Castro
- Correia
- Felipe
- Martins
- Melo

Problema do pertencimento

Solução imediata:

Criar uma estrutura de dados (*Set* / *HashSet*) com todos os elementos.

Proposta alternativa:

- Amanda Melo Araujo
- Carolina Correia Castro
- Felipe Martins Castro
- Nicolas Gomes Ferreira

- Carolina Correia Castro
- Igor Melo Cardoso
- Felipe Correia Araujo

- Amanda
- Araujo
- Carolina
- Castro
- Correia
- Felipe
- Martins
- Melo

Problema do pertencimento

Solução imediata:

Criar uma estrutura de dados (*Set* / *HashSet*) com todos os elementos.

Proposta alternativa:

- Amanda Melo Araujo
- Carolina Correia Castro
- Felipe Martins Castro
- Nicolas Gomes Ferreira

- Carolina Correia Castro
- Igor Melo Cardoso
- Felipe Correia Araujo

- Amanda
- Araujo
- Carolina
- Castro
- Correia
- Felipe
- Martins
- Melo

Estrutura *Bloom Filter*

Estrutura

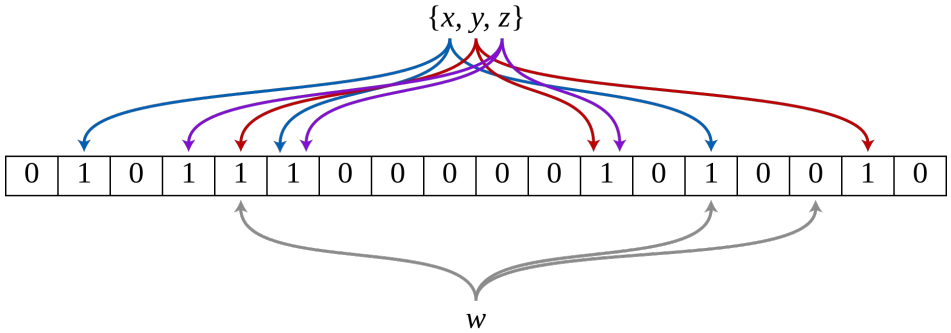
Um ***array* de *booleans***.

Inserção

Para cada ***ID***, calcule ***k*** *hashes*. Depois marque as posições relativas aos *hashes* calculados para ***True***.

Teste

Para cada ***ID***, calcule ***k*** *hashes.* Depois verifique se as posições são todas ***True***. Se sim retorne *True,* senão *False*.



Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome	Sobrenome

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome	Sobrenome
• Amanda (1)	• Melo (1)

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome	Sobrenome
• Amanda (1) • Carolina (1)	• Castro (1) • Melo (1)

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome	Sobrenome
• Amanda (1) • Carolina (2)	• Castro (2) • Melo (1)

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome	Sobrenome
• Amanda (1) • Carolina (2) • Felipe (1)	• Castro (3) • Melo (1)

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome	Sobrenome
Amanda (1)	Castro (3)
Carolina (2)	Melo (1)
Felipe (1)	

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome

- Amanda (1)
- Carolina (2)
- Felipe (1)

Sobrenome

- Castro (3)
- Melo (1)

Problema da frequência

Solução imediata:

Criar um *Map*/*HashMap* com os elementos, iniciando em 1, e incrementando a cada inserção. Caso não esteja no *Map*, retorna 0.

Proposta alternativa:

- Amanda Melo
- Carolina Castro
- Carolina Castro
- Felipe Castro

- Carolina Castro
- Igor Melo
- Felipe Melo

Nome

- Amanda (1)
- Carolina (2)
- Felipe (1)

Sobrenome

- Castro (3)
- Melo (1)

Estrutura
Count-Min Sketch

Estrutura

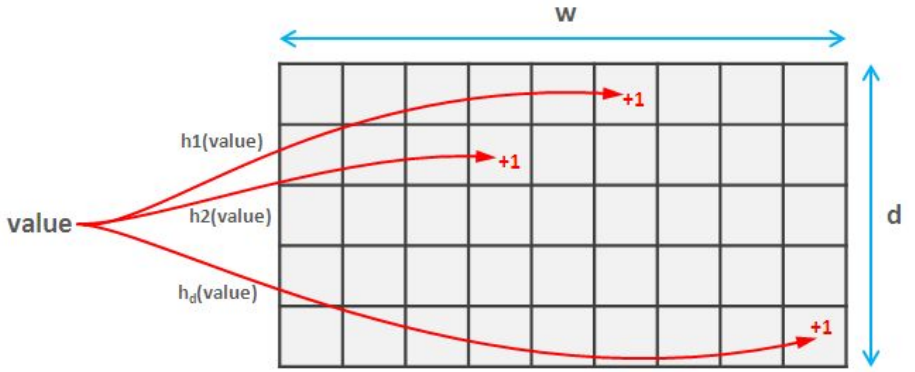
Uma ***matriz* de *inteiros***, onde cada linha é aplicado um *hash* diferente e a coluna é a indexação que o *hash* gerou.

Inserção

Para cada ***ID***, calcule ***k*** *hashes*, depois incremente em 1 o valor na célula relativa a cada *hash*.

Contagem

Para cada ***ID***, calcule ***k*** *hashes.* depois retorne o valor mínimo obtido nas células referenciadas pelas *hashes*.



Problema da cardinalidade

Solução imediata:

Criar uma estrutura de dados (*Set* / *HashSet*) com os elementos, e contar.

Proposta alternativa:

- 78169-4636
- 12004-4029
- 99941-4470
- 69798-0039
- 06023-9603
- 14515-9652
- 09588-4898
- ...

Problema da cardinalidade

Solução imediata:
Criar uma estrutura de dados (*Set* / *HashSet*) com os elementos, e contar.

Proposta alternativa:

- 78169-4603
- 12004-4029
- 99941-4470
- 69798-0039
- 06023-9636
- 14515-9652
- 09588-4898
- ...

bucket
0
2
3
6
8
9

Problema da cardinalidade

Solução imediata:
Criar uma estrutura de dados (*Set* / *HashSet*) com os elementos, e contar.

Proposta alternativa:

- 78169-4603
- 12004-4029
- 99941-4470
- 69798-0039
- 06023-9636
- 14515-9652
- 09588-4898
- ...

bucket	*max*
0	0
2	0
3	1
6	0
8	0
9	0

Problema da cardinalidade

Solução imediata:
Criar uma estrutura de dados (*Set* / *HashSet*) com os elementos, e contar.

Proposta alternativa:

- 78169-4603
- 12004-4029
- 99941-4470
- 69798-0039
- 06023-9636
- 14515-9652
- 09588-4898
- ...

bucket	*max*
0	0
2	0
3	1
6	0
8	0
9	0

$10 * 10^{mean(max)}$

Estrutura *HyperLogLog*

Estrutura

Um ***array* de *inteiros***.

Inserção

Para cada ***ID***, calcule ***um*** *hash*, use parte dele para definir o ***bucket***, e parte dele para contar os **zeros à direita**, salve no bucket o valor **máximo** entre o atual e o antigo.

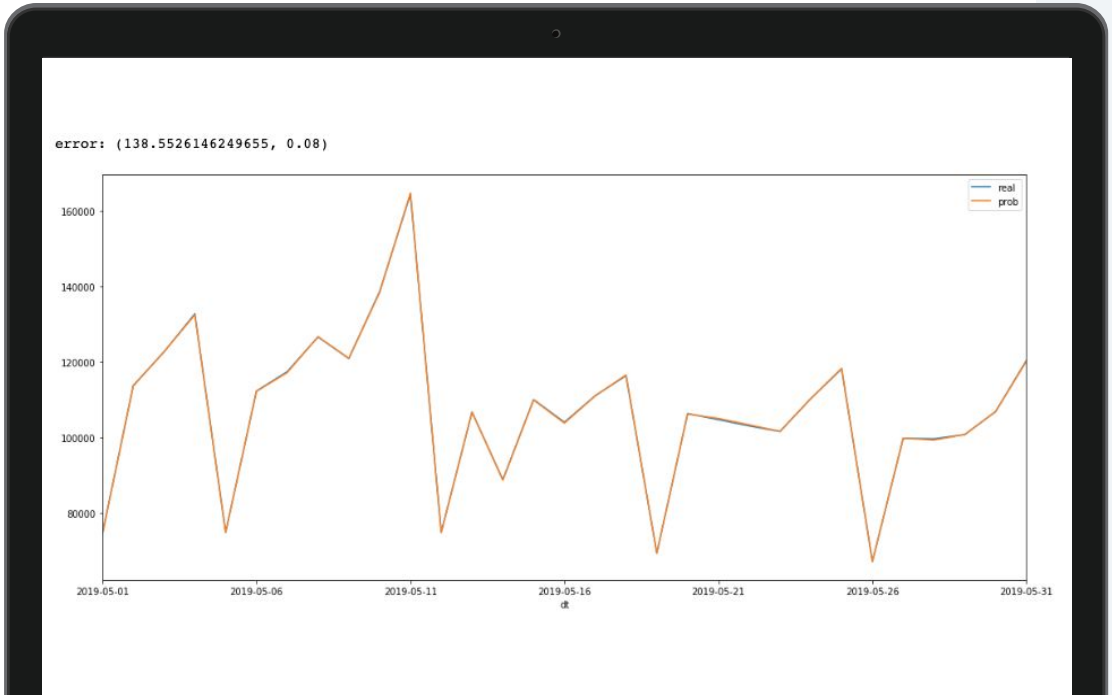
Agregação

Para cada **bucket** das estruturas, guarde o **máximo**.

Contagem

$$alpha * (n^o\ of\ buckets) * harm^{(2*maxs)}$$

Aplicações



Filtragem de contextos para redução de problemas de busca. (*Bloom Filter*)

Detecção de comportamentos anormais e ataques. (*Count-Min Sketch*)

Extração de métricas e consumo de contagens. (*HyperLogLog*)

Redução de tempo e memória em cerca de

99,99%

Agora que todos os dados estão em estruturas probabilísticas, elas estão privadas?

Agora que todos os dados estão em estruturas probabilísticas, elas estão privadas?

NÃO!

Agora que todos os dados estão em estruturas probabilísticas, elas estão privadas?

NÃO!

A invasão dos intrusos!

Usuários fictícios são criados aleatoriamente com probabilidade ***p***.

Agora que todos os dados estão em estruturas probabilísticas, elas estão privadas?

NÃO!

A invasão dos intrusos!

Usuários fictícios são criados aleatoriamente com probabilidade ***p***.

Os guardas da fronteira!

Usuários reais são impedidos de serem adicionados com probabilidade ***p***.

Agora que todos os dados estão em estruturas probabilísticas, elas estão privadas?

Não, mas podem ser!

A invasão dos intrusos!

Usuários fictícios são criados aleatoriamente com probabilidade ***p***.

Os guardas da fronteira!

Usuários reais são impedidos de serem adicionados com probabilidade ***p***.

inloco

lucas.rufino@inloco.com.br

inloco.com.br/careers
medium.com/inlocotech