

Orthogonal Matching Pursuit over Spark for Audio Compression

Tito Spadini

Inteligência na Web
Prof. Dr. Fabricio Olivetti de França

Maio 2018

Base de Dados

- Banco de Dados (≈ 100.000 trechos de áudio)
- Arquivos .wav
- Mono (1 canal)
- 16 bits
- 44.100 Hz
- Duração de 100 ms

Algoritmo

- Orthogonal Matching Pursuit (OMP)
- Algoritmo ganancioso
- Aprendizado baseado em Dicionários Esparsos

$$\mathcal{S}_{(m \times 1)} = \mathcal{D}_{(m \times n)} \times \mathcal{C}_{(n \times 1)}$$

sendo $\mathcal{S}_{(m \times 1)}$ o sinal; $\mathcal{D}_{(m \times n)}$, o Dicionário; e $\mathcal{C}_{(n \times 1)}$, o vetor de coeficientes esparsos.

Paralelizando funções-chave

- Transposta de matriz
- Multiplicação de matrizes
- Least Squares

Problema: desempenho terrivelmente ruim!

Paralelizando o Dicionário

Problemas:

- Melhora quase insignificante no desempenho
- Em vez de considerar todo o dicionário, passou a considerar apenas partes, o que influencia negativamente no comportamento do algoritmo OMP
- Não escalável

Paralelizando os Sinais

- Redução de 60% no tempo de execução!
- É escalável!

Conclusões

- Paralelizar o próprio OMP pode não ser eficiente
- Mapear os sinais com o uso do OMP parece ser uma solução melhor

Perguntas

