TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN
VIỆN TRÍ TUỆ NHÂN TẠO
-----***-----**

BÁO CÁO MÔN KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN

ĐỀ TÀI

THUẬT TOÁN K-MEANS & LẬP TRÌNH MAPREDUCE HÓA TRONG PHÂN CỤM ẢNH

Sinh viên thực hiện: Bùi Quang Vinh
Giảng viên hướng dẫn: TS. Trần Hồng Việt

Hà Nội, 12/2024

LỜI MỞ ĐẦU

Trong kỷ nguyên dữ liệu lớn, việc xử lý và phân tích khối lượng dữ liệu khổng lồ với tốc độ nhanh và độ chính xác cao đã trở thành một thách thức lớn. Các công nghệ và kỹ thuật dữ liệu lớn như Hadoop và Spark không chỉ cung cấp khả năng lưu trữ phân tán mà còn hỗ trợ xử lý dữ liệu ở quy mô rộng thông qua các mô hình lập trình song song như MapReduce. Trong lĩnh vực xử lý ảnh, phân cụm đóng vai trò then chốt trong nhận diện mẫu, phân đoạn hình ảnh và giảm nhiễu. Khi kết hợp thuật toán K-Means với MapReduce, chúng ta có thể tận dụng sức mạnh của các hệ thống dữ liệu lớn để tối ưu hóa hiệu suất tính toán và khả năng mở rộng.

Báo cáo này sẽ phân tích việc áp dụng hai công nghệ trên trong phân cụm ảnh, minh họa qua việc xử lý phân đoạn ảnh hàng không (semantic segmentation) nhằm nhận diện, phân biệt các đối tượng trong ảnh. Báo cáo được cấu trúc thành 5 phần:

Phần 1: Tổng quan về dữ liệu lớn
Phần 2: Giải thuật K-Means
Phần 3: Phân cụm ảnh sử dụng giải thuật K-Means song song MapReduce
Phần 4: Thí nghiệm
Phần 5: Kết luận

MỤC LỤC

PHẦN 1: TỔNG QUAN VỀ DỮ LIỆU LỚN VÀ MAPREDUCE

1.1. Định nghĩa Dữ liệu lớn

Dữ liệu lớn (Big Data) là thuật ngữ chỉ việc xử lý các tập hợp dữ liệu có quy mô lớn và phức tạp, vượt ngoài khả năng xử lý của các ứng dụng dữ liệu truyền thống. Dữ liệu lớn bao gồm các thách thức về phân tích, thu thập, giám sát, tìm kiếm, chia sẻ, lưu trữ, truyền tải, trực quan hóa, truy vấn và đảm bảo tính riêng tư của dữ liệu. Những thách thức này yêu cầu các phương pháp và công nghệ xử lý tiên tiến để khai thác giá trị từ dữ liệu.

Đặc trưng cơ bản của dữ liệu lớn:

1. **Khối lượng lớn (Volume):** Dữ liệu lớn có khối lượng rất lớn và không ngừng gia tăng. Tính đến năm 2014, dữ liệu có thể đạt đến hàng trăm terabyte, và xu hướng này tiếp tục gia tăng theo thời gian.
2. **Tốc độ (Velocity):** Dữ liệu được sinh ra và cập nhật với tốc độ cực nhanh, yêu cầu các hệ thống phải xử lý và phân tích dữ liệu ngay lập tức để đáp ứng nhu cầu thực tế.
3. **Đa dạng (Variety):** Hiện nay, hơn 80% dữ liệu được sinh ra là dữ liệu phi cấu trúc, bao gồm các loại dữ liệu như tài liệu văn bản, blog, hình ảnh, video, và âm thanh, tạo ra thách thức trong việc tổ chức và phân tích.
4. **Độ tin cậy/chính xác (Veracity):** Dữ liệu lớn thường chứa nhiều yếu tố nhiễu và không chính xác. Việc xử lý và làm sạch dữ liệu trở thành một nhiệm vụ quan trọng để đảm bảo độ tin cậy của kết quả phân tích.
5. **Giá trị (Value):** Mặc dù dữ liệu lớn có khối lượng và sự phức tạp cao, nhưng nếu được xử lý đúng cách, chúng có thể mang lại giá trị to lớn thông qua những thông tin quan trọng giúp ra quyết định và cải thiện hiệu suất.

1.2. Tổng quan về MapReduce

MapReduce là một mô hình lập trình song song, được Google phát triển để xử lý lượng dữ liệu lớn trong môi trường phân tán. Được thiết kế để giải quyết các vấn đề tính toán phân tán trên quy mô rộng, MapReduce giúp phân phối công việc tính toán đến hàng nghìn node trong một cụm máy tính, xử lý hàng petabyte dữ liệu một cách hiệu quả. Dự án Hadoop, cung cấp hệ thống tệp phân tán Hadoop

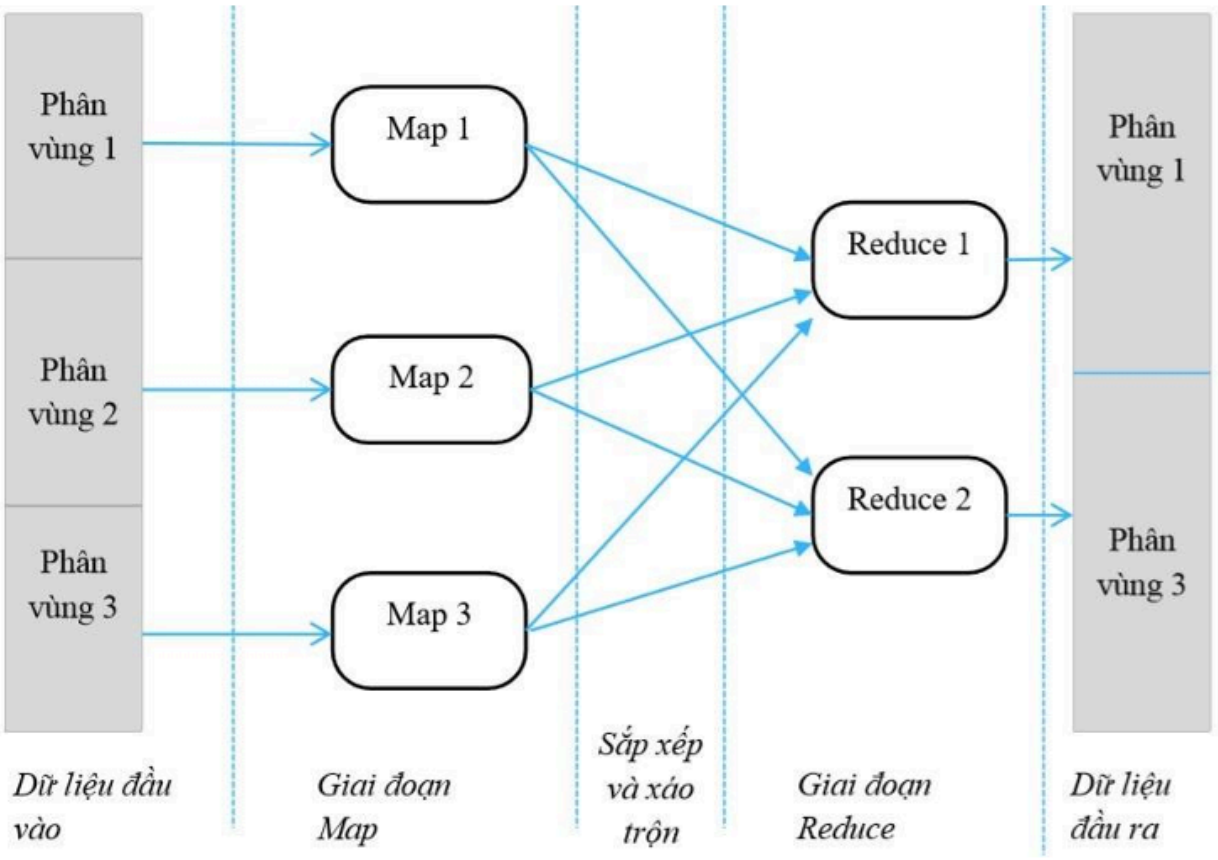
Distributed File System (HDFS), hỗ trợ mô hình MapReduce, cho phép lưu trữ và xử lý dữ liệu lớn trên quy mô phân tán.

MapReduce giúp xử lý các tác vụ tính toán yêu cầu khả năng mở rộng và song song hóa. Mỗi công việc trong MapReduce có thể xử lý từ hàng terabytes đến petabytes dữ liệu trên các node trong hệ thống phân tán. Dữ liệu đầu vào được chia thành các mảnh nhỏ và phân phối cho các node trong mạng. Mỗi node thực hiện một phần công việc, và số lượng cũng như kích thước của các mảnh dữ liệu này phụ thuộc vào số lượng node có sẵn trong hệ thống.

Quy trình thực hiện công việc trên MapReduce:

- **Bước 1:** Chia dữ liệu đầu vào thành các mảnh nhỏ: Dữ liệu được phân chia thành các phần nhỏ hơn, gọi là các mảnh, để có thể xử lý song song trên các node trong cụm.
- **Bước 2:** Thực hiện công việc Map trên từng mảnh dữ liệu đầu vào (xử lý song song trên nhiều máy tính trong cụm).
    - **Bộ ánh xạ (Mapper):** Xử lý tập dữ liệu đầu vào dưới dạng (key, value) và tạo ra tập dữ liệu trung gian với cặp (key, value).
        1. Ánh xạ dữ liệu đầu vào dưới dạng (key, value).
        2. Thực thi công việc Map để xử lý cặp (key, value) và tạo ra (key, value) mới, công việc này được gọi là chia nhóm.
        3. Kết quả đầu ra của bộ ánh xạ sẽ được lưu trữ và chuyển tới bộ giảm (Reducer) tương ứng.
- Bước 3: Tổng hợp kết quả trung gian (sắp xếp và trộn): Sau khi công việc Map hoàn tất, dữ liệu trung gian được sắp xếp và trộn lại để chuẩn bị cho công việc Reduce.
- Bước 4: Sau khi các cặp (key, value) trung gian đã được tổng hợp, công việc Reduce sẽ được thực hiện để xử lý các cặp dữ liệu này. Công việc này được thực hiện song song trên nhiều máy tính trong cụm.
- Bước 5: Tổng hợp kết quả từ hàm Reduce để cho ra kết quả cuối cùng.

Quá trình MapReduce cho phép thực hiện các tác vụ tính toán phức tạp trên lượng dữ liệu khổng lồ mà không cần phải xử lý tất cả dữ liệu trên một máy tính duy nhất, từ đó giúp tối ưu hóa hiệu suất và khả năng mở rộng khi làm việc với dữ liệu lớn.



Quy trình thực hiện công việc trên MapReduce

PHẦN 2: GIẢI THUẬT K-MEANS

2.1. Định nghĩa:

Phân cụm là một kỹ thuật quan trọng trong khai phá dữ liệu, thuộc nhóm các phương pháp học không giám sát trong học máy. Phân cụm được sử dụng để phân nhóm các đối tượng vào các cụm sao cho các đối tượng trong cùng một cụm có sự tương đồng cao, trong khi các đối tượng ở các cụm khác nhau lại có sự khác biệt lớn. Cụ thể, các đối tượng trong một cụm phải giống nhau về một số đặc điểm hoặc tính chất nhất định, còn các đối tượng ở các cụm khác thì khác nhau về những đặc điểm đó.

K-Means là một trong những thuật toán phân cụm phổ biến và quan trọng nhất trong kỹ thuật phân cụm. Mục tiêu chính của thuật toán K-Means là phân nhóm các đối tượng đã cho vào K cụm (K là số lượng cụm được xác định trước, là một số nguyên dương), sao cho tổng bình phương khoảng cách giữa các đối tượng và tâm cụm (centroid) là nhỏ nhất. Tức là, thuật toán sẽ tìm cách tối thiểu hóa tổng các khoảng cách từ các điểm dữ liệu đến trung tâm của các cụm mà chúng thuộc về.

2.2. Giải thuật K-Means:

Quá trình của thuật toán K-Means có thể được mô tả qua các bước cơ bản như sau:

1. **Khởi tạo các centroids ban đầu:** Đầu tiên, chọn **K điểm dữ liệu** ngẫu nhiên từ tập dữ liệu làm các centroid ban đầu (K là số cụm được xác định trước, và K < n, với n là số lượng điểm dữ liệu). Đây là bước quan trọng, vì nếu centroid được chọn không hợp lý, thuật toán có thể hội tụ vào kết quả không tối ưu.
2. Gán các điểm dữ liệu vào các cụm: Với mỗi điểm dữ liệu x_i, tính khoảng cách của nó đến tất cả các centroid C_j và gán điểm dữ liệu đó vào cụm có centroid gần nhất. Khoảng cách thường được tính bằng khoảng cách Euclid.
3. **Cập nhật các centroids:** Sau khi tất cả các điểm dữ liệu đã được gán vào các cụm, tính lại **tâm cụm mới** bằng cách lấy trung bình các điểm dữ liệu trong cụm đó. Cập nhật centroid của cụm j bằng công thức sau:

$$C_j = \frac{1}{|S_j|} \sum_{x \in S_j} x$$

4. **Lặp lại quá trình:**Các bước 2 và 3 được lặp lại cho đến khi một trong các điều kiện sau xảy ra:
    - Các centroids không thay đổi (hoặc thay đổi rất ít) so với lần cập nhật trước đó, tức là thuật toán đã hội tụ.
    - Số lần lặp đạt đến một ngưỡng tối đa được đặt ra từ trước.
    - Tổng lỗi (tổng bình phương khoảng cách từ các điểm dữ liệu đến centroid của cụm) không giảm đáng kể

Hạn chế của thuật toán K-Means:

- Trong trường hợp xấu nhất, độ phức tạp tính toán của K-Means có thể trở thành superpolynomial, đặc biệt khi số lượng điểm dữ liệu rất lớn và số cụm K cao. Tuy nhiên, với các trường hợp thực tế, thuật toán thường có độ phức tạp thời gian là O(K·n·t), trong đó:
    - K là số cụm,
    - n là số điểm dữ liệu,
    - t là số lần lặp.
- Việc khởi tạo các centroid ngẫu nhiên có thể khiến cho quá trình phân cụm trở nên khó khăn hơn. Các centroid ban đầu không hợp lý có thể dẫn đến kết quả phân cụm không

tối ưu và sự khác biệt trong kết quả mỗi lần chạy thuật toán. Để khắc phục điều này, có thể sử dụng phương pháp **K-Means++** để khởi tạo centroid sao cho chúng phân bố đều hơn, giúp cải thiện chất lượng phân cụm và giảm thiểu khả năng thuật toán hội tụ vào kết quả kém.

PHẦN 3: PHÂN CỤM ẢNH SỬ DỤNG GIẢI THUẬT K-MEANS SONG SONG VỚI MAPREDUCE

3.1. Bài toán:

Trong các ứng dụng phân tích ảnh, phân đoạn ảnh hàng không là một trong những bài toán quan trọng trong lĩnh vực xử lý ảnh địa lý (remote sensing).
Mục tiêu: phân chia ảnh hàng không thành các vùng (hoặc lớp) khác nhau dựa trên các đặc tính giống nhau của các đối tượng hoặc khu vực trong ảnh.
Giải pháp:
Để giải quyết bài toán phân đoạn ảnh hàng không, chúng ta có thể sử dụng giải thuật K-Means trong môi trường phân tán, cụ thể là sử dụng MapReduce để xử lý dữ liệu lớn và tăng tốc độ tính toán. Thuật toán K-Means sẽ được áp dụng để phân nhóm các pixel trong ảnh thành các cụm dựa trên các đặc tính của chúng.
K-means giúp chia tập dữ liệu (pixel ảnh) thành các cụm tương ứng với các vùng trong ảnh có đặc điểm tương đồng.
MapReduce hỗ trợ phân tán và song song hóa quá trình phân cụm, giúp xử lý các lượng dữ liệu lớn mà không gặp phải vấn đề về hiệu suất

3.2. Triển khai

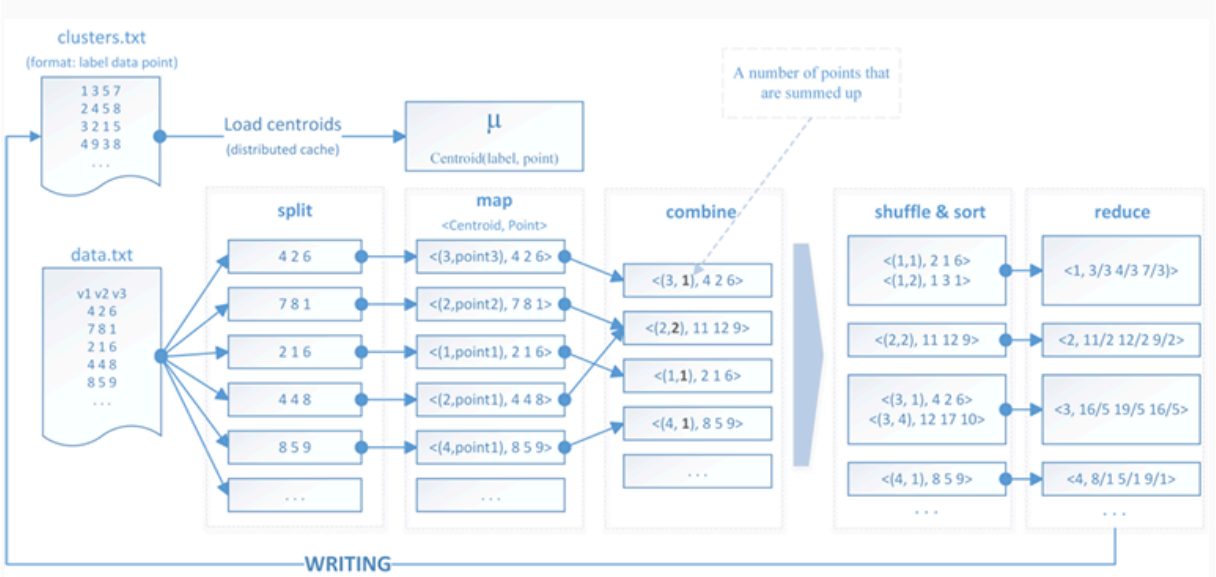
1. K-Means và K-Means++

- **K-Means++**: Chọn tâm cụm ban đầu ngẫu nhiên, sau đó sử dụng khoảng cách giữa các điểm để xác định các tâm cụm tiếp theo.
- **K-Means**: Gán điểm dữ liệu vào tâm cụm gần nhất và cập nhật các tâm cụm bằng trung bình của các điểm trong cụm, lặp lại đến khi hội tụ.

2. Quá Trình MapReduce

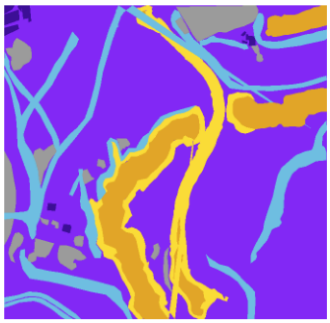
- **MapReduce - Map Phase**:
  - Chia nhỏ tệp dữ liệu đầu vào.
  - Mỗi điểm dữ liệu được xử lý bởi hàm map.
  - Gắn mỗi pixel với centroid gần nhất và tạo các cặp key-value <Centroid, Point>.
- **MapReduce - Combiner**:
  - Giảm số lượng ghi cục bộ.
  - Cộng lại các điểm dữ liệu trên cùng một máy và ghi lại số lượng.
- **MapReduce - Shuffle and Sort**:
  - Các giá trị đầu ra được xáo trộn và sắp xếp theo Centroid.
- **MapReduce - Reduce Phase**:

- Cập nhật centroid của mỗi cụm.
- Kiểm tra điều kiện hội tụ giữa các centroid cũ và mới.
- Nếu centroid hội tụ, dừng lại; nếu không, tăng Counter và chạy lại MapReduce.



3. Kiểm Tra Điều Kiện và Xử Lý Kết Quả

- **Kiểm tra điều kiện hoàn thành**:
  - Chương trình hoàn tất nếu số lần lặp tối đa được đạt hoặc nếu Counter không thay đổi.
- **Xử lý Sau Phân Cụm**:
  - Làm sáng các pixel trong cụm liên quan đến phần đất dành cho xây dựng.
  - Làm tối phần còn lại của ảnh.
- **Kết quả Đầu Ra**:
  - Tạo ảnh nhị phân với phần đất dành cho xây dựng được làm sáng (màu trắng) và phần còn lại tối (màu đen). ảnh masking là ảnh có sẵn để so sánh hiệu suất.

Ảnh gốc	Ảnh masking	Ảnh thu được
		

PHẦN 4: THÍ NGHIỆM VÀ KẾT QUẢ

4.1. Thí nghiệm

Dataset:

- **Bộ dữ liệu gồm 16 ảnh chụp vệ tinh**: Bộ dữ liệu bao gồm 16 ảnh chụp vệ tinh, mỗi ảnh chứa các khu vực có thể xây dựng (được chỉ ra bằng màu trắng) và các khu vực còn lại (được chỉ ra bằng màu đen).
- **Bộ dữ liệu gồm 16 ảnh mask**: Bộ dữ liệu này bao gồm 16 ảnh nhị phân, trong đó các vùng trắng đại diện cho khu vực có thể xây dựng, còn phần đen là các khu vực còn lại.

So sánh kết quả phân cụm với ảnh mask:

- Các ảnh phân đoạn được chuyển thành ảnh nhị phân (với vùng trắng đại diện cho khu vực có thể xây dựng).
- Sử dụng **Dice Similarity Coefficient (DSC)** để so sánh ảnh phân đoạn với ảnh mask, đánh giá mức độ chính xác của thuật toán phân cụm.
- Công thức tính Dice Similarity Coefficient là:

$$\text{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

Trong đó:

- A là tập hợp các điểm ảnh trắng trong ảnh phân đoạn
- B là tập hợp các điểm ảnh trắng sau khi nhị phân hóa trong ảnh mask
- $|A \cap B|$ là số điểm ảnh chung giữa hai ảnh.
- $|A|$ và $|B|$ là tổng số điểm ảnh trong mỗi ảnh.

Lưu và phân tích kết quả:

- Kết quả so sánh của mỗi cặp ảnh sẽ được lưu trong tệp **comparison_results.txt**.
- Các chỉ số Dice được tính toán cho từng cặp ảnh, từ đó đánh giá mức độ chính xác của phân cụm.
- Cuối cùng, độ giống nhau cao nhất và trung bình của tất cả các ảnh sẽ được tính toán và hiển thị.

Kết quả

- **comparison_results.txt** chứa kết quả so sánh của 16 cặp ảnh. Kết quả phân tích cho thấy độ giống nhau cao nhất giữa ảnh phân đoạn và ảnh mask là 87.41%, độ giống nhau trung bình là 67.15%, chứng minh hiệu quả của thuật toán phân cụm KMeans trong việc xác định các khu vực có thể xây dựng từ ảnh vệ tinh.

4.2. Đánh giá

Ưu điểm

- Việc kết hợp với MapReduce giúp xử lý dữ liệu lớn trên cụm máy tính, rất phù hợp khi cần phân cụm trên lượng lớn ảnh vệ tinh có độ phân giải cao. Mặc dù trong bài toán này chỉ sử dụng 16 ảnh, MapReduce có thể mở rộng hiệu quả khi số lượng ảnh tăng lên.
- MapReduce chia nhỏ công việc thành các tác vụ nhỏ hơn (map), đồng thời thực hiện giai đoạn giảm (reduce) để tổng hợp kết quả, giúp tối ưu hóa thời gian xử lý, đặc biệt khi có nhiều ảnh vệ tinh cần phân cụm.
- Việc tính toán tâm cụm và phân cụm lại có thể được tự động hóa qua các vòng lặp MapReduce, giúp giảm bớt công sức thủ công khi xử lý với dữ liệu lớn.
- Thuật toán K-means kết hợp với MapReduce rất dễ triển khai trong môi trường phân tán, giúp xử lý nhanh chóng và mở rộng quy mô khi cần thiết.

Hạn chế

- Mặc dù K-means có thể phân cụm các khu vực có thể xây dựng và không xây dựng, nhưng các ảnh vệ tinh có thể chứa nhiều chi tiết phức tạp, như nhiễu hoặc nhiều loại hình khác nhau, khiến thuật toán khó phân biệt chính xác giữa các khu vực có thể xây dựng và các khu vực còn lại nếu không có bước tiền xử lý kỹ càng.
- Thuật toán K-means chỉ xem xét các giá trị pixel (màu sắc và độ sáng) mà không tận dụng ngữ cảnh không gian hoặc đặc điểm hình học của các khu vực trong ảnh, điều này có thể dẫn đến lỗi phân cụm, đặc biệt khi các khu vực xây dựng không đồng nhất về hình dạng.
- Khi xử lý lượng dữ liệu nhỏ (chỉ 16 ảnh trong bài toán này), MapReduce có thể tạo ra chi phí lớn hơn lợi ích do quá trình khởi tạo và quản lý các tác vụ phân tán, làm giảm hiệu suất trong trường hợp số lượng ảnh ít.

Đề xuất cải tiến

- Cải thiện chất lượng ảnh vệ tinh trước khi áp dụng phân cụm. Các bước tiền xử lý như lọc nhiễu (Gaussian, Median), cân bằng độ sáng hoặc làm sắc nét ảnh có thể giúp tăng độ chính xác trong việc phân biệt các khu vực xây dựng và không xây dựng.
- **Tối ưu hóa thuật toán K-means**:
    - Để cải thiện chất lượng phân cụm, có thể sử dụng thuật toán K-means++ thay vì việc khởi tạo các centroid một cách ngẫu nhiên. K-means++ giúp cải thiện sự phân bố của các centroid ban đầu, giảm thiểu khả năng rơi vào các địa phương tối ưu cục bộ.
    - Ngoài K-means, các thuật toán phân cụm như DBSCAN hoặc thuật toán phân cụm dựa trên đặc trưng không gian có thể được thử nghiệm để tận dụng các đặc điểm hình học của các khu vực trong ảnh.
- Để cải thiện khả năng phân cụm, có thể áp dụng các kỹ thuật học sâu như mạng nơ-ron tích chập (CNN) để khai thác các đặc trưng không gian của ảnh, giúp cải thiện độ chính xác trong việc phân biệt các khu vực xây dựng và không xây dựng.

PHẦN 5: KẾT LUẬN

Trong báo cáo này, chúng ta đã phân tích và triển khai thuật toán K-Means kết hợp với công nghệ MapReduce để giải quyết bài toán phân cụm ảnh, đặc biệt là trong việc phân đoạn ảnh hàng không nhằm nhận diện và phân biệt các khu vực có thể xây dựng từ ảnh vệ tinh. Việc sử dụng MapReduce đã giúp phân tán và song song hóa quá trình xử lý, tối ưu hóa hiệu suất và khả năng mở rộng khi làm việc với dữ liệu lớn.

Kết quả thực nghiệm cho thấy thuật toán K-Means kết hợp với MapReduce có thể phân cụm hiệu quả và chính xác, với độ tương đồng trung bình đạt được 67.15% so với các ảnh mask thực tế. Điều này chứng tỏ khả năng của phương pháp này trong việc xử lý và phân tích ảnh vệ tinh, đặc biệt trong các bài toán phân loại vùng đất có thể xây dựng.

Tuy nhiên, vẫn còn một số hạn chế cần khắc phục, đặc biệt là khi xử lý các ảnh có nhiều nhiễu hoặc các khu vực có hình dạng phức tạp, điều này có thể ảnh hưởng đến độ chính xác của thuật toán. Trong các trường hợp này, việc áp dụng các phương pháp tiền xử lý dữ liệu như lọc nhiễu hoặc cải thiện thuật toán K-Means có thể giúp nâng cao kết quả.

Với khả năng mở rộng cao và hiệu suất tính toán được tối ưu hóa nhờ MapReduce, phương pháp này có thể được áp dụng trong các dự án phân tích ảnh vệ tinh quy mô lớn, giúp nâng cao hiệu quả công việc trong các ứng dụng thực tế như quy hoạch đô thị, giám sát đất đai, và bảo vệ môi trường.

Trong tương lai, có thể cải thiện kết quả phân cụm bằng cách kết hợp thêm các phương pháp học sâu hoặc các thuật toán phân cụm khác như DBSCAN hoặc phương pháp học máy giám sát để cải thiện độ chính xác và khả năng xử lý các tình huống phức tạp hơn.

Tài liệu tham khảo

(1) https://github.com/markomih/kmeans_mapreduce/tree/master -
(2) https://github.com/niamyaraghi/Intro-to-ML/blob/main/Color%20Segmentation%20(clustering).ipynb