TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VIỆN TRÍ TUỆ NHÂN TẠO

-----*-----**

BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN

ĐỀ TÀI

K-Means và Lập trình MapReduce trong phân cụm ảnh

Nhóm sinh viên thực hiện:

1. Nguyễn Phú Lộc : 22022547
2. Phạm Chiến : 22022634
3. Bùi Ngọc Khánh : 22022551
4. Nguyễn Quang Huy : 22022582

Giảng viên hướng dẫn: TS. Trần Hồng Việt

HÀ NỘI, 12/2024

MỤC LỤC

MỞ ĐẦU

Công nghệ Big Data đang ngày càng khẳng định vai trò quan trọng trong việc thúc đẩy sự phát triển của các ngành công nghiệp hiện đại. Với khả năng lưu trữ, phân tích và xử lý khối lượng dữ liệu khổng lồ, Big Data đã trở thành nền tảng cốt lõi cho việc ra quyết định dựa trên dữ liệu trong nhiều lĩnh vực như y tế, tài chính, giao thông, giáo dục, và nhiều lĩnh vực khác.

Kể từ khi đạt được vị trí quan trọng trong bảng xếp hạng các công nghệ mới nổi của Gartner vào tháng 8/2015, Big Data đã tạo nên một cuộc cách mạng, mở ra những cơ hội lớn cho việc giải quyết các thách thức phức tạp trong thực tế. Để hiện thực hóa tiềm năng của Big Data, các framework xử lý dữ liệu lớn, đặc biệt là Hadoop - với khả năng lưu trữ phân tán và áp dụng mô hình lập trình MapReduce - đã được phát triển mạnh mẽ và đóng vai trò then chốt.

Trong bối cảnh đó, kỹ thuật phân cụm dữ liệu, đặc biệt là thuật toán K-Means, được xem là một trong những công cụ hiệu quả để khai thác giá trị từ dữ liệu lớn. Đây là lý do nhóm chúng em quyết định chọn đề tài: *"K-Means và Lập trình MapReduce trong phân cụm ảnh"* để làm báo cáo kết thúc môn học của mình. Thông qua đề tài này, chúng em mong muốn tìm hiểu sâu hơn về cách ứng dụng thực tiễn của Big Data và các công cụ xử lý dữ liệu lớn, từ đó góp phần nâng cao kiến thức và kỹ năng trong lĩnh vực công nghệ dữ liệu.

Báo cáo gồm 4 chương:

Chương 1: Tổng quan về dữ liệu lớn.

Chương 2: Thuật toán K-means

Chương 3: Ứng dụng MapReduce trong phân cụm ảnh bằng K-Means

Chương 4: Kết luận và hướng phát triển.

Đường dẫn tới repository: https://github.com/phulocnguyen/KMeans-MapReduce-Images-Clustering

I. Tổng quan về dữ liệu lớn

1. Định nghĩa dữ liệu lớn

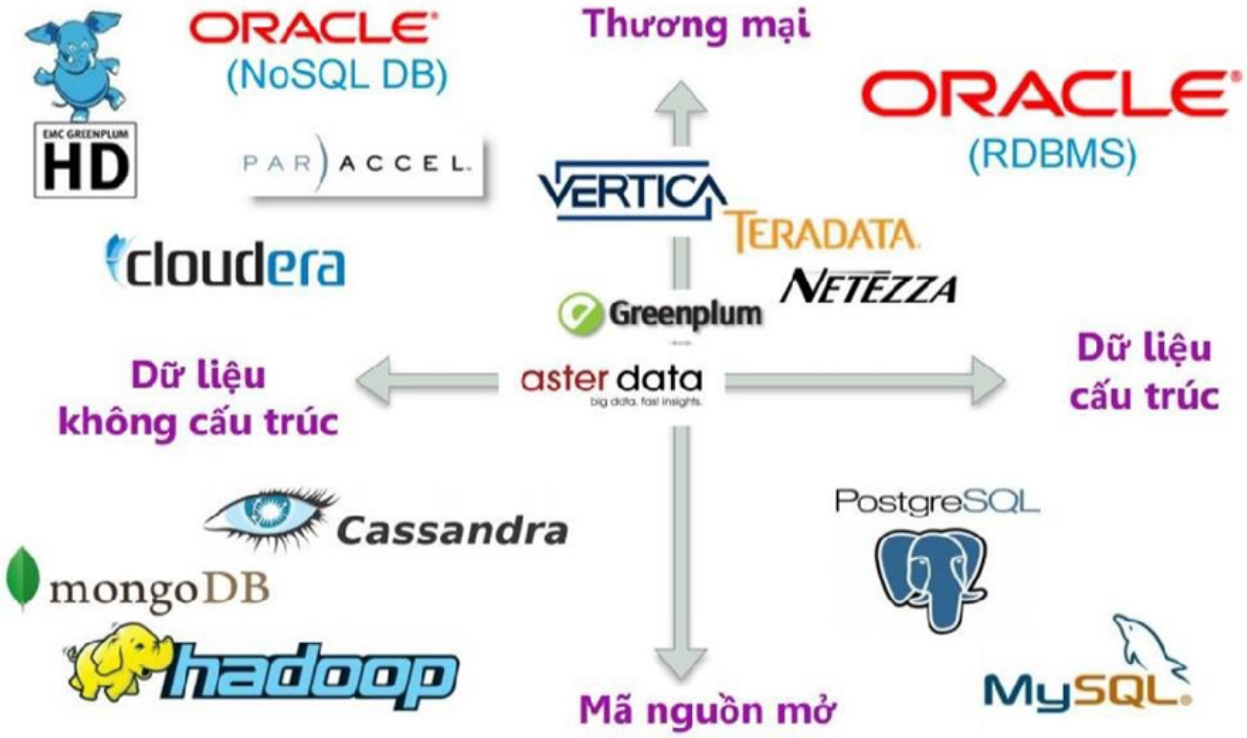
- Theo Wikipedia: Dữ liệu lớn là bộ dữ liệu có kích thước hoặc độ phức tạp lớn đến mức các phương pháp xử lý truyền thống không đáp ứng được.

- **Theo Gartner**: Dữ liệu lớn được đặc trưng bởi **3Vs**:

+ **Volume** (Khối lượng lớn): Lượng dữ liệu khổng lồ, tăng trưởng nhanh qua thời gian.

+ **Velocity** (Tốc độ): Tốc độ tạo ra và xử lý dữ liệu rất nhanh.

+ **Variety** (Đa dạng): Dữ liệu đến từ nhiều nguồn và có nhiều định dạng khác nhau



2. Đặc trưng cơ bản của dữ liệu lớn

- **Volume** (Khối lượng lớn): Lượng dữ liệu khổng lồ, tăng trưởng nhanh qua thời gian.

- **Velocity** (Tốc độ): Tốc độ tạo ra và xử lý dữ liệu rất nhanh.

- **Variety** (Đa dạng): Dữ liệu đến từ nhiều nguồn và có nhiều định dạng khác nhau.

- **Veracity** (Độ chính xác): Chất lượng và độ tin cậy của dữ liệu.

- **Value** (Giá trị): Lợi ích thu được từ dữ liệu.

3. Công nghệ chính trong xử lý dữ liệu lớn

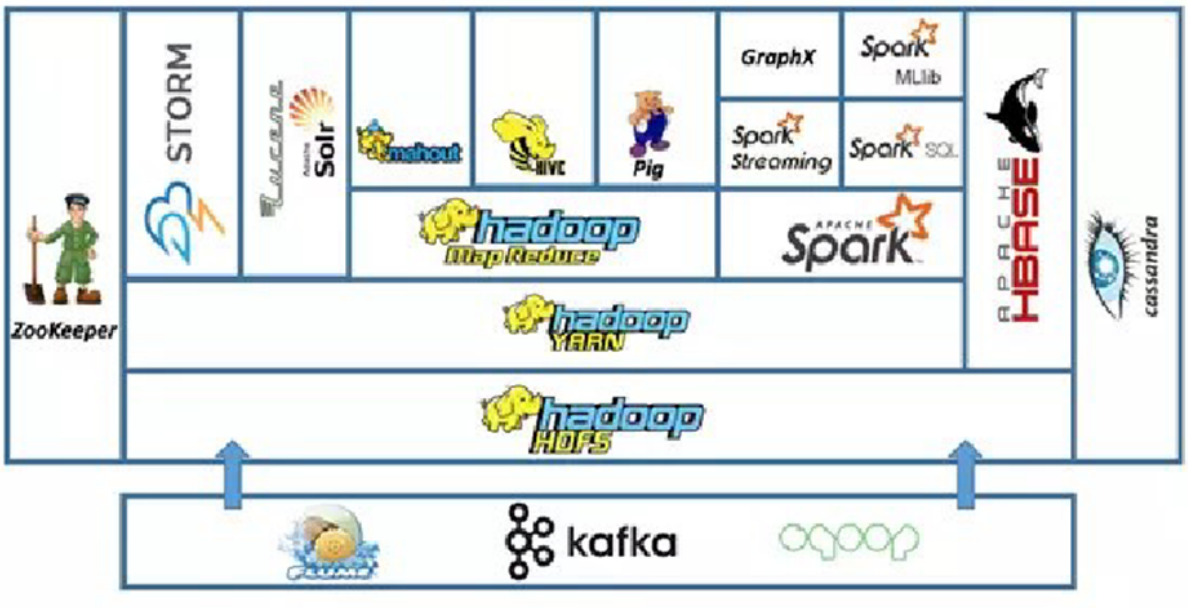
- Tính toán phân tán

- Tính toán song song

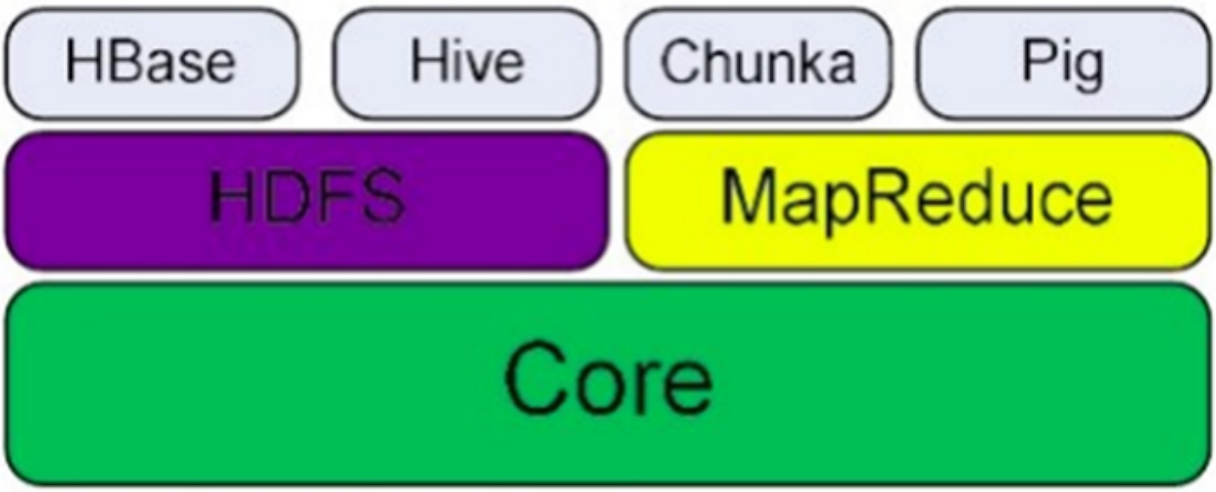
- Song song hóa bằng CPU đa nhân
- Song song hóa bằng GPU
- Xử lý phân tán với hệ thống cluster
- Xử lý phân tán trên Cloud

4. Tổng quan hadoop

Hadoop là một công nghệ phân tán và mã nguồn mở được sử dụng phổ biến để xử lý và lưu trữ khối dữ liệu lớn trên các cụm máy tính phân tán, được thiết kế để xử lý và lưu trữ dữ liệu lớn một cách hiệu quả. Hadoop được tạo ra bởi Doug Cutting và Mike Cafarella và năm 2005, và được phát triển bởi Apache Software Foundation dựa trên công nghệ Google File System và MapReduce. Cung cấp khả năng xử lý dữ liệu lớn, mở rộng linh hoạt và chi phí thấp, làm cho nó trở thành một công nghệ không thể thiếu trong các hệ thống xử lý dữ liệu hiện đại. Nó được phát triển để giải quyết các thách thức trong lĩnh vực Big Data mà các công nghệ cũ không thể đáp ứng.



- Các thành phần của hadoop: Core, MapReduce engine, HDFS, HBase, Hive, Pig, Chukwa,.. Tuy nhiên tập chung vào 2 thành phần quan trọng nhất: HDFS và MapReduce.



- Hadoop cho phép xử lý dữ liệu theo lô và có khả năng xử lý khối lượng dữ liệu cực lớn. Hadoop sử dụng một cụm các máy tính (server) thông thường để lưu trữ, tính toán. Việc tính toán này trên HDFS được thực hiện một cách song song, đồng thời và trừu tượng với các lập trình viên giúp họ tránh được việc lập trình mạng và xử lý bài toán đồng bộ phức tạp. Không giống như nhiều hệ thống phân tán khác, Hadoop cung cấp việc xử lý logic trên nơi lưu trữ dữ liệu mà không phải lấy dữ liệu từ các máy khác giúp tăng hiệu năng một cách mạnh mẽ.

- Hadoop bao gồm nhiều module như:

+ Hadoop Common: các tiện ích cơ bản hỗ trợ Hadoop.

+ Hadoop Distributes File System (HDFS): Hệ thống file phân tán cung cấp khả năng truy vấn song song tối đa hóa theo đường truyền truy cập bởi ứng dụng.

+ Hadoop YARN: Framework quản lý lập lịch tác vụ và quản lý các tài nguyên trên cụm.

+ Hadoop MapReduce: Hệ thống YARN-based để xử lý tập dữ liệu lớn

Kết luận: Là một framework cho phép phát triển các ứng dụng phân tán. Viết bằng java.

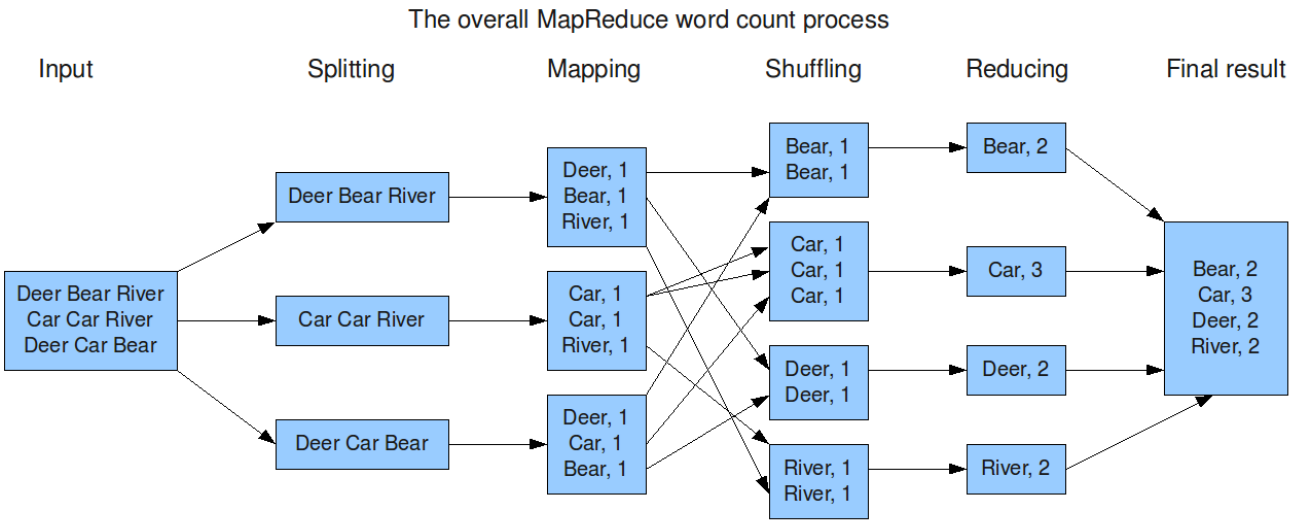
5. Tổng quan MapReduce

MapReduce là một khung làm việc xử lý dữ liệu phân tán được sử dụng để xử lý dữ liệu lớn trên Hadoop. MapReduce được phát triển bởi Google và sau đó được Apache Software Foundation phát triển và phát hành dưới dạng một phần của hệ sinh thái Hadoop.

MapReduce thực hiện xử lý dữ liệu bằng cách phân tách dữ liệu thành các phần nhỏ hơn và xử lý chúng song song trên các nút máy tính khác nhau trong cùng một mạng Hadoop. Khung làm việc MapReduce được thiết kế để hoạt động trên các phần dữ liệu độc lập, do đó, các phần dữ liệu có thể được xử lý độc lập và đồng thời tăng tốc độ xử lý.

MapReduce bao gồm hai pha chính là pha Map và pha Reduce. Trong pha Map, dữ liệu được xử lý và phân tích bằng các hàm Map để tạo ra các cặp key-value. Key-value này sau đó

được chuyển đến pha Reduce để được tổng hợp và xử lý tiếp theo. Trong pha Reduce, các cặp key-value được tổng hợp và xử lý bằng các hàm Reduce để tạo ra kết quả cuối cùng.

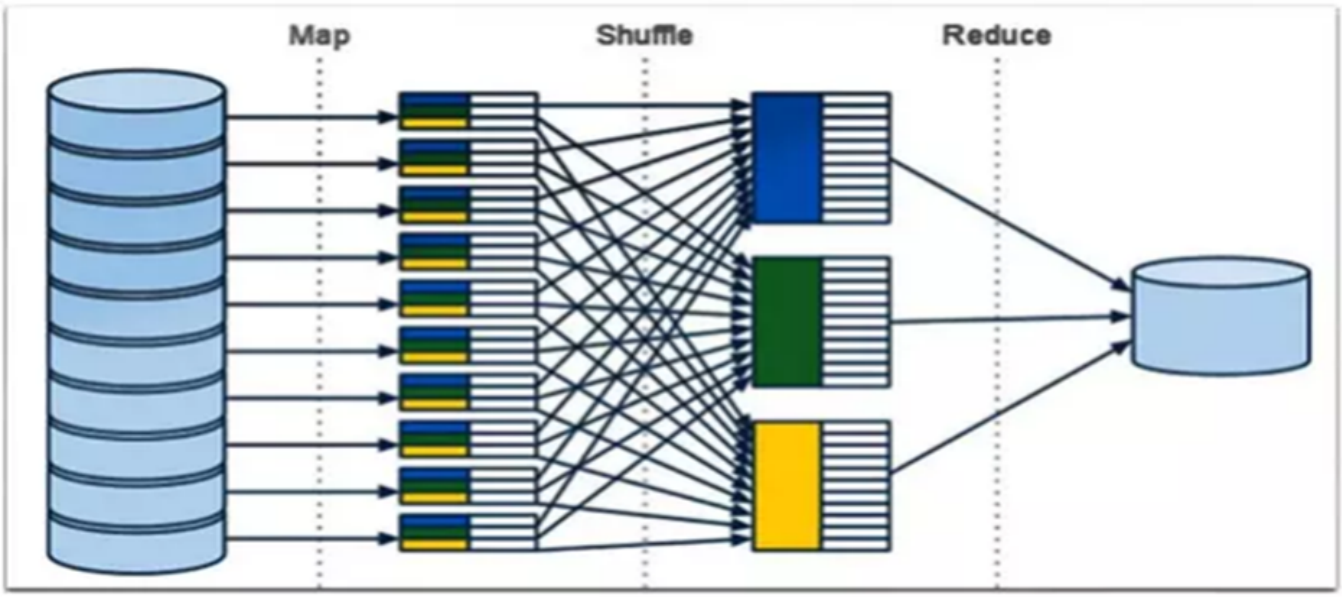


- Các bước của MapReduce:

+ Bước Map: Dữ liệu vào được chia thành các phần nhỏ hơn và được xử lý độc lập trên từng nút trong cụm Hadoop.

+ Bước Shuffle: Dữ liệu đầu ra từ bước Map được sắp xếp và gom nhóm để chuẩn bị cho bước Reduce.

+ Bước Reduce: Dữ liệu được xử lý lại trên các nút trong cụm Hadoop và kết quả cuối cùng được trả về.



- Lợi ích của MapReduce:

+ Có thể xử lý các tập dữ liệu lớn và phức tạp trên nhiều máy chủ trong một cụm.

+ Đạt được hiệu suất và tốc độ xử lý cao bằng cách phân tách các tác vụ xử lý thành các công việc nhỏ hơn và thực hiện trên nhiều máy chủ đồng thời.

+ Đảm bảo tính độc lập và bất biến của các công việc xử lý, do đó giảm thiểu tác động của lỗi trong quá trình xử lý dữ liệu.

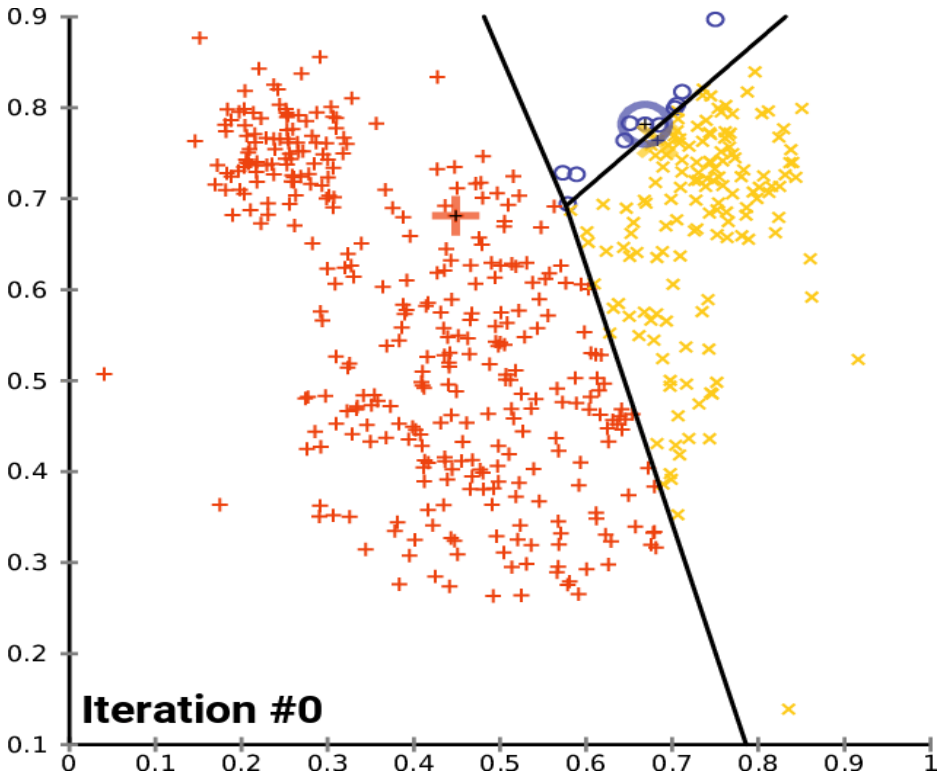
+ Có thể mở rộng hệ thống để xử lý tập dữ liệu lớn hơn hoặc tăng tốc độ xử lý bằng cách thêm nhiều máy chủ vào cụm Hadoop

II. Thuật Toán K-Means Và Ứng Dụng K-Means Trong Phân Cụm Ảnh

1. Thuật toán K-means

1.1 Thuật toán K-Means là gì?

Theo Wikipedia ***Phân cụm K-Means*** là một phương pháp lượng tử hóa vectơ , có nguồn gốc từ xử lý tín hiệu , nhằm mục đích phân chia *n* quan sát thành *k* cụm trong đó mỗi quan sát thuộc về cụm có giá trị trung bình gần nhất (tâm cụm hoặc trọng tâm cụm), đóng vai trò là nguyên mẫu của cụm.



K-means là một thuật toán học máy không giám sát , nhóm các tập dữ liệu không có nhãn thành các cụm khác nhau. Học máy không giám sát là quá trình dạy máy tính sử dụng dữ

liệu không có nhãn, không được phân loại và cho phép thuật toán hoạt động trên dữ liệu đó mà không cần giám sát. Không có bất kỳ quá trình huấn luyện dữ liệu nào trước đó, công việc của máy trong trường hợp này là sắp xếp dữ liệu chưa được sắp xếp theo các điểm song song, mẫu và biến thể.

K-Means có nghĩa là phân cụm, gán các điểm dữ liệu cho một trong K cụm tùy thuộc vào khoảng cách của chúng từ tâm của các cụm. Nó bắt đầu bằng cách gán ngẫu nhiên các cụm tâm trong không gian. Sau đó, mỗi điểm dữ liệu được gán cho một trong các cụm dựa trên khoảng cách của nó từ tâm của cụm. Sau khi gán mỗi điểm cho một trong các cụm, các cụm tâm mới được gán. Quá trình này chạy lặp đi lặp lại cho đến khi tìm thấy cụm tốt. Trong phân tích, chúng ta giả định rằng số lượng cụm được đưa ra trước và chúng ta phải đặt các điểm vào một trong các nhóm.

Ưu điểm

- Phân cụm K-Means hoạt động tốt nhất khi dữ liệu được phân tách tốt
- K-Means nhanh hơn so với các kỹ thuật phân cụm khác. Nó cung cấp sự kết hợp mạnh mẽ giữa các điểm dữ liệu.

Nhược điểm:

- Trong một số trường hợp, K không được định nghĩa rõ ràng và chúng ta phải nghĩ về số lượng K tối ưu.
- Tuy nhiên, khi các điểm dữ liệu chồng chéo nhau, phân cụm này không phù hợp.
- Phân cụm K-Means không cung cấp thông tin rõ ràng về chất lượng của các cụm. Việc chỉ định ban đầu khác nhau của trọng tâm cụm có thể dẫn đến các cụm khác nhau.
- Ngoài ra, thuật toán K-Means nhạy cảm với nhiễu. Nó có thể bị kẹt ở các cực tiểu cục bộ.

1.2 Lịch sử phát triển

Theo Wikipedia thuật ngữ "k-means" được James MacQueen sử dụng đầu tiên vào năm 1967, mặc dù ý tưởng này có từ Hugo Steinhaus vào năm 1956.

Thuật toán chuẩn lần đầu tiên được Stuart Lloyd của Bell Labs đề xuất vào năm 1957 như một kỹ thuật điều chế mã xung , mặc dù nó không được công bố dưới dạng bài báo trên tạp chí cho đến năm 1982

Năm 1965, Edward W. Forgy đã công bố về cơ bản cùng một phương pháp, đó là lý do tại sao nó đôi khi được gọi là thuật toán Lloyd–Forgy.

1.3 Mô tả thuật toán

Cho một tập hợp các quan sát ($x_1, x_2,...,x_n$), trong đó mỗi quan sát là một vectơ thực d chiều,

cụm K-Means nhằm mục đích phân chia n quan sát thành k (k ≤ n) tập S = $\{S_1, S_2, \ldots, S_k\}$ để giảm tổng bình phương trong cụm (hay là phương sai). Thực chất là tìm:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Trong đó μ_i là giá trị trung bình của các điểm S_i tức là:

$$\boldsymbol{\mu_i} = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}$$

$|S_i|$ là kích thước của S_i và || . || là chuẩn L^2 thông thường. Điều này tương đương với việc giảm độ lệch bình phương từng cặp của các điểm trong cùng một cụm:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2$$

Sự tương đương có thể suy ra từ $|S_i| \sum_{x,y \in S_i} ||x - \mu_1||^2 = \frac{1}{2} \sum_{x,y \in S_i} ||x - y||^2$. Vì tổng phương sai là hằng số, điều này tương đương với việc tối đa hóa tổng độ lệch bình phương giữa các điểm trong các cụm khác nhau (tổng bình phương giữa các cụm). Mối quan hệ xác định này cũng liên quan đến quy luật tổng phương sai trong lý thuyết xác suất.

1.4 Hoạt động của thuật toán

Chúng ta được cung cấp một tập dữ liệu các mục, với các tính năng nhất định và các giá trị cho các tính năng này (như một vectơ). Nhiệm vụ là phân loại các mục đó thành các nhóm. Để đạt được điều này, chúng ta sẽ sử dụng thuật toán K-means, một thuật toán học không giám sát. 'K' trong tên của thuật toán biểu thị số nhóm/cụm mà chúng ta muốn phân loại các mục của mình thành.

(Sẽ hữu ích nếu bạn nghĩ về các mục như các điểm trong không gian n chiều). Thuật toán sẽ phân loại các mục thành k nhóm hoặc cụm có độ tương đồng. Để tính độ tương đồng đó, chúng ta sẽ sử dụng khoảng cách Euclidean làm phép đo.

Thuật toán k-means có thể chia thành các bước như sau:

Bước 1: Tạo các trung tâm ngẫu nhiên

$$\mathbb{C}^{(0)} = \left\{ m_1^{(0)},\ m_2^{(0)},\ \ldots,\ m_k^{(0)} \right\}$$

Bước 2: Gán các điểm dữ liệu vào các cụm

Với mỗi điểm dữ liệu, ta sẽ tính khoảng cách của nó tới các trung tâm (bằng khoảng

cách Euclid). Ta sẽ gán chúng vào trung tâm gần nhất. Tập hợp các điểm được gán vào cùng 1 trung tâm sẽ tạo thành cụm.

$$\mathbb{S}_i^{(t)} = \left\{ x_p : \| x_p - m_i^{(t)} \|^2 \leq \| x_p - m_j^{(t)} \|^2 \right\}, \forall j,\ 1 \leq j \leq k$$

Bước 3: Cập nhật trung tâm

Với mỗi cụm đã tìm được ở bước 2, trung tâm mới sẽ là trung bình cộng của các điểm dữ liệu trong cụm đó.

$$m_i^{(t+1)} = \frac{1}{\left| \mathbb{S}_i^{(t)} \right|} \sum_{x \in \mathbb{S}_i^{(t)}} x_j$$

Thuật toán sẽ lặp lại các bước trên cho tới khi đạt được kết quả chấp nhận được.

Các "điểm" được đề cập ở trên được gọi là trung bình vì chúng là giá trị trung bình của các mục được phân loại trong đó. Để khởi tạo các trung bình này, chúng ta có nhiều tùy chọn. Một phương pháp trực quan là khởi tạo các trung bình tại các mục ngẫu nhiên trong tập dữ liệu. Một phương pháp khác là khởi tạo các trung bình tại các giá trị ngẫu nhiên giữa các ranh giới của tập dữ liệu (nếu đối với một tính năng *x,* các mục có giá trị trong [0,3], chúng ta sẽ khởi tạo các trung bình với các giá trị cho *x* tại [0,3]).

1.5 Ứng dụng

Phân đoạn hình ảnh : K-Means có thể được sử dụng để phân đoạn hình ảnh thành các vùng khác nhau dựa trên các đặc điểm màu sắc hoặc kết cấu của chúng. Điều này hữu ích trong các ứng dụng thị giác máy tính, chẳng hạn như nhận dạng hoặc theo dõi đối tượng.

Phân khúc khách hàng : K-Means có thể được sử dụng để nhóm khách hàng thành các phân khúc khác nhau dựa trên thói quen mua sắm, dữ liệu nhân khẩu học hoặc các đặc điểm khác. Điều này hữu ích trong các ứng dụng tiếp thị và quảng cáo, vì nó có thể giúp các doanh nghiệp nhắm mục tiêu các nỗ lực tiếp thị của họ hiệu quả hơn.

Phát hiện bất thường: K-Means có thể được sử dụng để xác định các giá trị ngoại lệ hoặc bất thường trong một tập dữ liệu. Điều này hữu ích trong việc phát hiện gian lận, phát hiện xâm nhập mạng và các ứng dụng bảo mật khác.

Phân cụm tài liệu: K-Means có thể được sử dụng để nhóm các tài liệu tương tự lại với nhau dựa trên nội dung của chúng. Điều này hữu ích trong các ứng dụng xử lý ngôn ngữ tự nhiên, chẳng hạn như phân loại văn bản hoặc phân tích tình cảm.

Hệ thống đề xuất : K-Means có thể được sử dụng để đề xuất sản phẩm hoặc dịch vụ cho người dùng dựa trên các giao dịch mua hoặc sở thích trước đây của họ. Điều này hữu ích trong các

ứng dụng thương mại điện tử và quảng cáo trực tuyến.

2. Ứng dụng K-means trong phân cụm ảnh

2.1 Giới thiệu

Phân vùng ảnh là một quá trình chia một bức ảnh thành nhiều vùng khác nhau, mỗi vùng gồm các pixel có tính tương đồng cao và tách biệt với các vùng lân cận.

K-means là một thuật toán phân cụm không giám sát, được sử dụng rộng rãi trong phân vùng ảnh nhờ vào tính đơn giản và hiệu quả cao.

2.2. Các phép đo khoảng cách

2.2.1 Khoảng cách Euclidean

Khoảng cách Euclid (Euclidean Distance) là một khái niệm trong toán học dùng để đo khoảng cách "trực tiếp" giữa hai điểm trong không gian Euclid. Đây là một dạng mở rộng của định lý Pythagoras trong không gian nhiều chiều.

Khoảng cách Euclid giữa hai điểm $x = (x_1, x_2, \dots, x_n)$và $y = (y_1, y_2, \dots, y_n)$ trong không gian n chiều được tính bằng công thức:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

2.2.2. Khoảng cách Manhattan

Khoảng cách Manhattan (Manhattan Distance) hay còn gọi là khoảng cách L1, là một phương pháp đo khoảng cách giữa hai điểm trong không gian bằng cách cộng tổng giá trị tuyệt đối của sự khác biệt giữa các tọa độ tương ứng của chúng.

Khái niệm này xuất phát từ việc đi lại trên một lưới đường phố dạng hình chữ nhật, giống như các đường phố ở Manhattan, New York.

Khoảng cách Manhattan giữa hai điểm $x = (x_1, x_2, \dots, x_n)$và $y = (y_1, y_2, \dots, y_n)$ trong không gian n chiều được tính bằng công thức:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

2.2.3. Khoảng cách Cosine

Khoảng cách Cosine (Cosine Distance) là một phương pháp đo lường sự khác biệt giữa hai vector trong không gian đa chiều. Thay vì tập trung vào độ lớn của vector, khoảng cách Cosine chủ yếu quan tâm đến góc giữa các vector, thể hiện độ tương đồng hoặc không tương đồng về hướng.

Khoảng cách Cosine được tính dựa trên **Cosine Similarity**. Nếu hai vector u và v có góc giữa chúng là θ, thì Cosine Similarity được định nghĩa là:

$$\text{Cosine Similarity} = \cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$

$$\text{Cosine Distance} = 1 - \cos(\theta)$$

2.2.4. Khoảng cách Minkowski

Khoảng cách Minkowski là một cách tổng quát hóa của các khoảng cách phổ biến như **khoảng cách Euclid** và **khoảng cách Manhattan**. Nó được sử dụng trong không gian đa chiều để đo lường khoảng cách giữa hai điểm dựa trên một tham số ppp, cho phép điều chỉnh mức độ ảnh hưởng của từng thành phần.

Khoảng cách Minkowski giữa hai điểm $u = (u_1, u_2, \dots, u_n)$và $v = (v_1, v, \dots, v_n)$ trong không gian n chiều được tính bằng công thức:

$$d(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{n} |u_i - v_i|^p \right)^{\frac{1}{p}}$$

2.3 Nguyên lý hoạt động K-Means trong phân cụm ảnh

2.3.1. K-Means trong phân cụm các điểm ảnh

- **Nguyên tắc**: Dựa vào khoảng cách giữa các điểm ảnh

- **Khởi tạo**
    - Xác định số lượng cụm (được gọi là k).
    - Chọn ngẫu nhiên k tâm cụm ban đầu.
- **Phân cụm các điểm ảnh**
    - Tính khoảng cách Euclid (Manhattan, Cosine, Minkowski) giữa các điểm ảnh và tâm cụm.
    - Gán pixel vào cụm gần nhất.
- **Cập nhật tâm cụm**
    - Tính trung bình toạ độ cho các pixel trong cùng cụm và đặt làm tâm cụm mới.
- **Lặp lại**
    - Tiếp tục phân nhóm và cập nhật tâm cụm cho đến khi hội tụ.

2.3.2. K-Means trong phân cụm các ảnh có cùng đặc điểm

- **Nguyên tắc**: Dựa vào độ tương đồng giữa các vector Embedding của các ảnh
- **Khởi tạo**
    - Xác định số lượng cụm (được gọi là k).
    - Chọn ngẫu nhiên k tâm cụm ban đầu.
- **Phân cụm các điểm ảnh**
    - Tính khoảng cách Euclid (Manhattan, Cosine, Minkowski) giữa các vector Embedding và tâm cụm.
    - Gán ảnh vào cụm gần nhất.
- **Cập nhật tâm cụm**
    - Cập nhật tâm cụm bằng trung bình của các vector Embedding được gán cho cụm ngay tại lần lặp đó
- **Lặp lại**
    - Tiếp tục phân nhóm và cập nhật tâm cụm cho đến khi hội tụ.

2.3 Ưu điểm, nhược điểm

Ưu điểm

- Tính đơn giản, dễ hiểu và triển khai nhanh.
- Khả năng xử lý lượng dữ liệu lớn (ảnh có độ phân giải lớn, số ảnh lớn)

Nhược điểm:

- Nhạy cảm với việc khởi tạo tâm cụm.

- Có thể hội tụ vào cực tiểu cục bộ nếu không khởi tạo tốt.
- Hiệu quả phụ thuộc vào việc chọn số cụm k.
- Độ chính xác không cao bằng các phương pháp hiện đại

2.4 So sánh với các phương pháp phân vùng khác

Tiêu chí	K-means	Fuzzy C-means	Threhold-base
Tính đơn giản	Cao	Trung bình	Cao
Tính linh hoạt	Thấp (đối với biến đổi cạnh)	Cao	Cao
Độ nhạy với	Cao	Thấp	Trung bình

2.5 Kết luận

- Thuật toán K-Means là công cụ hữu hiệu cho phân vùng ảnh nhờ tính đơn giản và hiệu quả cao. Tuy nhiên, cần khắc phục nhược điểm về việc khởi tạo và chọn tham số k để đạt được kết quả tối ưu.
- Trong tương lai, có thể kết hợp với các phương pháp khác như subtractive clustering để cải thiện độ chính xác và hiệu năng.

III. Ứng dụng MapReduce trong phân cụm ảnh bằng K-Means

1. **Ý tưởng KMeans dựa trên MapReduce**

Khởi tạo tâm cụm (centroids):

Tâm cụm ban đầu được khởi tạo ngẫu nhiên (hoặc sử dụng một phương pháp khác như KMeans++). Những tâm này sẽ được chia sẻ giữa các node trong hệ thống phân tán.

Giai đoạn Map:

Mỗi mapper nhận một phần dữ liệu từ bộ dữ liệu lớn.

Với mỗi điểm dữ liệu trong phần đó, mapper xác định tâm cụm gần nhất (dựa trên khoảng cách Euclidean hoặc các phương pháp tính khoảng cách khác).

Mapper tạo ra cặp giá trị dưới dạng: `(tâm cụm, điểm dữ liệu)`.

Giai đoạn Shuffle and Sort:

Hệ thống MapReduce sẽ nhóm tất cả các điểm dữ liệu thuộc cùng một tâm cụm lại với nhau. Điều này tạo ra các nhóm điểm dữ liệu tương ứng với mỗi tâm cụm.

Giai đoạn Reduce:

Mỗi reducer nhận nhóm các điểm dữ liệu thuộc về một tâm cụm.

Reducer tính toán tâm cụm mới bằng cách lấy trung bình tọa độ của tất cả các điểm dữ liệu trong nhóm.

Tâm cụm mới sẽ được gửi lại để khởi chạy bước lặp tiếp theo.

Lặp lại (Iterative Process):

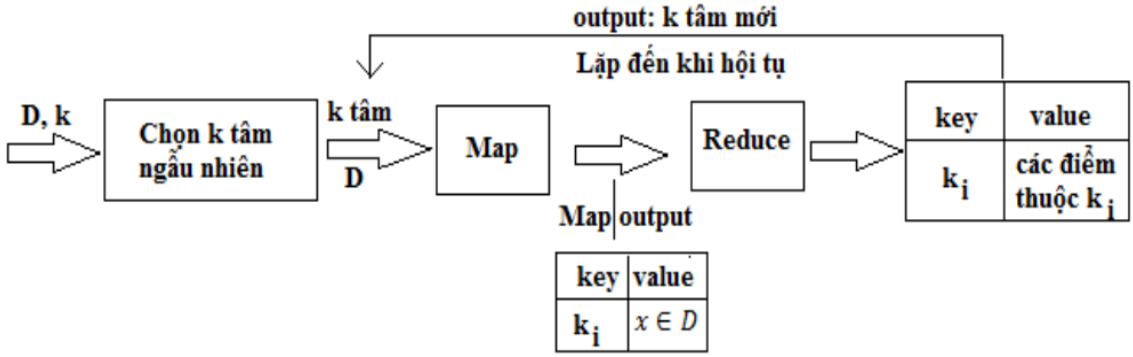
Các tâm cụm mới được cập nhật và phân phối đến các mapper để tiếp tục xử lý.

Quy trình này được lặp lại cho đến khi các tâm cụm hội tụ (tức là không thay đổi đáng kể giữa các bước lặp) hoặc đạt đến số lần lặp tối đa.

Kết quả:

Sau khi hội tụ, các tâm cụm cuối cùng được lưu trữ, và mỗi điểm dữ liệu sẽ được gán nhãn dựa trên tâm cụm gần nhất.

2. Lưu đồ thuật toán



2.1. Dữ liệu đầu vào

- Dữ liệu đầu vào là toạ độ điểm ảnh hoặc tập hợp ảnh được biểu diễn dưới dạng vector đặc trưng
- Ví dụ $x_i = [f_1, f_2, \ldots, f_n]$ trong đó f_k là gí trị đặc trưng thứ k của ảnh

2.2. Kmeans Mapper

Input:

- keyIn: Số thứ tự ảnh
- valIn: Vector đặc trưng của ảnh

Output:

- keyInt:: Chỉ số cụm j mà ảnh thuộc về.
- valInt:Vector đặc trưng của ảnh

2.3. KMeans Reducer

Input:

- keyIn: Chỉ số cụm j
- valIn: Danh sách các vector đặc trưng của cụm j

Output:

- keyOut: Chỉ số cụm j
- valOut:Tâm cụm mới $c_j{}^{t+1}$

2.4. Điều kiện hội tụ

Thuật toán ngừng lặp khi đạt đến điều kiện hội tụ :

$$||c_j{}^{t+1} - c_j{}^{t}||_2 \leq \delta$$

3. Phân cụm các điểm ảnh (Image Compression)

- **Dữ liệu**: Một bức ảnh bất kì được biểu diễn dưới dạng các điểm ảnh
- **Thử nghiệm**: Thực hiện phân cụm các điểm ảnh với nhiều tham số khác nhau bao gồm số cluster, phép đo khoảng cách khác nhau (Euclid, Cosine, Manhattan, …)
- **Kết quả**: Thuật toán hội tụ sau 7 lần chạy
-

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus
application_1733393643055_0024	phulocnguyen	kmeans_mapreduce.jar	MAPREDUCE		root.default	0	Wed Dec 11 15:01:45 +0700 2024	Wed Dec 11 15:01:49 +0700 2024	Wed Dec 11 15:02:05 +0700 2024	FINISHED	SUCCEEDED

```
Counter value = 1
Algorithm converged!
Checking path /KMeans/resources/output/6/part-r-[0-9]*
Adding hdfs://localhost:9000/KMeans/resources/output/6/part-r-00000
Adding hdfs://localhost:9000/KMeans/resources/output/6/part-r-00001
Adding hdfs://localhost:9000/KMeans/resources/output/6/part-r-00002
2024-12-11 15:01:44,281 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-12-11 15:01:44,297 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/phul
ocnguyen/.staging/job_1733393643055_0024
2024-12-11 15:01:44,777 INFO input.FileInputFormat: Total input files to process : 1
2024-12-11 15:01:44,829 INFO mapreduce.JobSubmitter: number of splits:1
2024-12-11 15:01:45,269 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1733393643055_0024
2024-12-11 15:01:45,270 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-12-11 15:01:45,286 INFO impl.YarnClientImpl: Submitted application application_1733393643055_0024
2024-12-11 15:01:45,289 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1733393643055_
0024/
2024-12-11 15:01:45,289 INFO mapreduce.Job: Running job: job_1733393643055_0024
2024-12-11 15:01:54,444 INFO mapreduce.Job: Job job_1733393643055_0024 running in uber mode : false
2024-12-11 15:01:54,444 INFO mapreduce.Job:  map 0% reduce 0%
2024-12-11 15:01:59,496 INFO mapreduce.Job:  map 100% reduce 0%
2024-12-11 15:02:03,530 INFO mapreduce.Job:  map 100% reduce 33%
2024-12-11 15:02:04,534 INFO mapreduce.Job:  map 100% reduce 67%
2024-12-11 15:02:05,540 INFO mapreduce.Job:  map 100% reduce 100%
2024-12-11 15:02:06,556 INFO mapreduce.Job: Job job_1733393643055_0024 completed successfully
2024-12-11 15:02:06,595 INFO mapreduce.Job: Counters: 51
```

```
Found 7 items
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 14:59 /KMeans/resources/output/1
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 15:00 /KMeans/resources/output/2
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 15:00 /KMeans/resources/output/3
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 15:00 /KMeans/resources/output/4
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 15:01 /KMeans/resources/output/5
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 15:01 /KMeans/resources/output/6
drwxr-xr-x   - phulocnguyen supergroup          0 2024-12-11 15:02 /KMeans/resources/output/7
```

4. Phân cụm các ảnh (Image Clustering)

- **Dữ liệu**: Sử dụng bộ dữ liệu nhỏ trên Kaggle dùng cho bài toán phân cụm “Small image dataset for unsupervised clustering”, bao gồm 80 ảnh thuộc 5 class khác nhau (dog, cat, family, alone, food)

- **Thử nghiệm**: Trước khi đưa vào mô hình, các ảnh được biểu diễn dưới dạng vector đặc trưng. Dữ liệu được huấn luyện với nhiều phép đo khoảng cách khác nhau
- **Kết quả**: Thuật toán hội tụ sau 10 lần chạy

```
Counter value = 2
Algorithm converged!
Checking path /KMeans/resources/output/5/part-r-[0-9]*
Adding hdfs://localhost:9000/KMeans/resources/output/5/part-r-00000
Adding hdfs://localhost:9000/KMeans/resources/output/5/part-r-00001
Adding hdfs://localhost:9000/KMeans/resources/output/5/part-r-00002
2024-12-11 15:01:21,935 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-12-11 15:01:21,952 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/phul
ocnguyen/.staging/job_1733393643055_0023
2024-12-11 15:01:22,028 INFO input.FileInputFormat: Total input files to process : 1
2024-12-11 15:01:22,492 INFO mapreduce.JobSubmitter: number of splits:1
2024-12-11 15:01:22,930 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1733393643055_0023
2024-12-11 15:01:22,931 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-12-11 15:01:22,952 INFO impl.YarnClientImpl: Submitted application application_1733393643055_0023
2024-12-11 15:01:22,955 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1733393643055_
0023/
2024-12-11 15:01:22,955 INFO mapreduce.Job: Running job: job_1733393643055_0023
2024-12-11 15:01:32,078 INFO mapreduce.Job: Job job_1733393643055_0023 running in uber mode : false
2024-12-11 15:01:32,078 INFO mapreduce.Job:  map 0% reduce 0%
2024-12-11 15:01:37,122 INFO mapreduce.Job:  map 100% reduce 0%
2024-12-11 15:01:41,161 INFO mapreduce.Job:  map 100% reduce 33%
2024-12-11 15:01:42,185 INFO mapreduce.Job:  map 100% reduce 67%
2024-12-11 15:01:43,194 INFO mapreduce.Job:  map 100% reduce 100%
2024-12-11 15:01:44,211 INFO mapreduce.Job: Job job_1733393643055_0023 completed successfully
```

IV. Kết luận và hướng phát triển

1. Kết luận

Đề tài "K-Means và Lập trình MapReduce trong phân cụm ảnh" đã tập trung khai thác ứng dụng của Big Data trong lĩnh vực xử lý và phân cụm ảnh. Qua quá trình nghiên cứu và thực hiện, nhóm chúng em đã đạt được những kết quả đáng khích lệ:

- Hiểu rõ khái niệm, vai trò của Big Data, cùng các công cụ xử lý dữ liệu lớn như Hadoop và mô hình MapReduce, đặc biệt là cách tối ưu hóa xử lý dữ liệu phân tán.
- Nắm bắt chi tiết thuật toán K-Means, các bước thực hiện, ưu nhược điểm, và các ứng

dụng thực tế trong phân cụm dữ liệu.

- Triển khai thành công chương trình kết hợp K-Means với MapReduce để thực hiện phân cụm ảnh, qua đó khai thác được tiềm năng của việc xử lý dữ liệu lớn trong việc phân tích hình ảnh .
- Thử nghiệm chương trình với tập dữ liệu ảnh cụ thể, thu được kết quả khả quan và tiến hành đánh giá hiệu năng của chương trình.

Tuy nhiên, bài báo cáo vẫn còn một số hạn chế cần khắc phục:

- Quy mô dữ liệu thử nghiệm còn nhỏ, chưa đủ để đánh giá toàn diện khả năng mở rộng và hiệu quả của hệ thống trong các bài toán thực tế với dữ liệu lớn.
- Chương trình demo còn hạn chế trong việc hỗ trợ các tập dữ liệu có tính đa dạng cao hoặc các định dạng phức tạp khác.
- Các chiến lược khởi tạo cụm và lựa chọn số cụm k chưa được tối ưu, dẫn đến việc phân cụm đôi khi không đạt hiệu quả cao nhất.

2. Hướng phát triển

Trong tương lai, nhóm chúng em định hướng mở rộng và nâng cao đề tài theo các hướng sau:

- Mở rộng quy mô xử lý: Phát triển chương trình để xử lý dữ liệu lớn hơn, ứng dụng vào các lĩnh vực thực tế như phân tích ảnh y tế, ảnh vệ tinh, hệ thống giám sát giao thông, và phân cụm dữ liệu mạng xã hội.
- Nâng cao độ chính xác: Kết hợp thuật toán K-Means với các phương pháp hiện đại như K-Means++, Gaussian Mixture Models (GMM), hoặc tích hợp với các mô hình học sâu để tăng cường khả năng nhận diện và phân cụm chính xác hơn.
- Cải thiện hiệu suất: Nghiên cứu và triển khai các phép đo khoảng cách đa dạng như Cosine, Mahalanobis, hoặc Minkowski để tăng tính linh hoạt của thuật toán khi làm việc với các tập dữ liệu có cấu trúc khác nhau.
- Tích hợp công nghệ mới: Áp dụng các công cụ và framework hiện đại như Apache Spark, TensorFlow hoặc PyTorch để xử lý dữ liệu nhanh hơn và hiệu quả hơn.
- Phát triển ứng dụng đa nền tảng: Tích hợp chương trình vào các hệ thống thực tế như ứng dụng di động hoặc nền tảng web, từ đó nâng cao tính khả dụng và phổ biến của giải pháp.
- Nghiên cứu thêm thuật toán mới: Khám phá và thử nghiệm các thuật toán phân cụm khác như DBSCAN, Spectral Clustering để đánh giá hiệu năng so với K-Means.

Nhóm chúng em hy vọng rằng những hướng phát triển trên không chỉ cải thiện chất lượng chương trình mà còn đóng góp vào sự phát triển của các ứng dụng Big Data trong thực tiễn. Rất mong nhận được sự góp ý từ thầy cô và các bạn để hoàn thiện hơn nữa báo cáo và kỹ năng của nhóm.

TÀI LIỆU THAM KHẢO

(1) Ghazal, Taher & Hussain, Muhammad & Said, Raed & Nadeem, Afrozah & Hasan, Mohammad Kamrul & Ahmad, Munir & Khan, Muhammad & Naseem, Muhammad & Muhammad, Adnan. (2021). Performances of K-Means Clustering Algorithm with Different Distance Metrics.

(2) Zhao, Weizhong & Ma, Huifang & He, Qing. (1970). Parallel K-Means Clustering Based on MapReduce. Cloud computing. 5931. 674-679. 10.1007/978-3-642-10665-1_71.

(3) Dang Thi Thu Hien - DS Lab (VIASM). Cluster Analysis Lecture

(4) Small image dataset for unsupervised clustering - Won Du Chang on Kaggle

(5) https://github.com/markomih/kmeans_mapreduce

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

Họ và tên	**Công việc**
Nguyễn Phú Lộc	- Tìm hiểu về mô hình K-means với MapReduce trong Image Compression - Lập trình và thử nghiệm với dữ liệu khác nhau - Viết báo cáo - Ý tưởng MapReduce hoá KMeans - Thử nghiệm K-Means với nhiều phép đo khoảng cách khác nhau
Phạm Chiến	- Tìm hiểu về mô hình K-means trong phân cụm ảnh - Lập trình và thử nghiệm - Viết báo cáo - Đánh giá và kết luận
Nguyễn Quang Huy	- Tìm hiểu về tổng quan của thuật toán K-Means trong phân cụm ảnh - Tìm nguồn dữ liệu để thử nghiệm - Viết báo cáo
Bùi Ngọc Khánh	- Viết báo cáo - Tìm hiểu về mô hình lập trình MapReduce - Tìm hiểu cách hoạt động của Kmeans với MapReduce