1. Introduction

Kính chào thầy cô và các bạn, hôm nay nhóm chúng em xin trình bày về **Principal Component Analysis (PCA)** – một phương pháp quan trọng trong xử lý dữ liệu và học máy.

📌 **Vấn đề đặt ra:** Trong thực tế, dữ liệu thường có **quá nhiều biến số (dimensions)**, làm cho việc phân tích trở nên khó khăn, mô hình học máy dễ bị quá khớp (overfitting) và tốn nhiều tài nguyên tính toán.

📌 **Giải pháp:** PCA là một kỹ thuật giúp **giảm số chiều dữ liệu**, nhưng vẫn giữ lại **lượng thông tin quan trọng nhất**, giúp dữ liệu dễ hiểu hơn và tăng hiệu quả tính toán.

📌 **Ứng dụng thực tế:**

✅ **Nhận diện khuôn mặt** trong trí tuệ nhân tạo

✅ **Giảm nhiễu** trong dữ liệu hình ảnh

✅ **Tăng tốc độ học máy** bằng cách giảm số lượng biến số

Trong bài thuyết trình này, chúng em sẽ giải thích **cách PCA hoạt động**, các **bước thực hiện**, và một số **ứng dụng thực tế** để mọi người hiểu rõ hơn về kỹ thuật này. Hy vọng bài trình bày sẽ mang lại nhiều thông tin hữu ích! 🚀

1. Purpose

Mục tiêu chính của PCA là **giảm số chiều của dữ liệu** trong khi vẫn giữ lại **phần lớn thông tin quan trọng**. Điều này giúp:

✅ **Đơn giản hóa dữ liệu**: Biểu diễn dữ liệu có nhiều biến dưới dạng một tập hợp nhỏ hơn nhưng vẫn giữ được xu hướng chính.

✅ **Giảm nhiễu và loại bỏ dư thừa**: PCA giúp loại bỏ các biến có mối tương quan cao, làm cho dữ liệu rõ ràng hơn.

✅ **Tăng hiệu suất mô hình học máy**: Giảm số lượng biến giúp mô hình học nhanh hơn, giảm nguy cơ quá khớp (overfitting).

✅ **Dễ dàng trực quan hóa dữ liệu**: Khi dữ liệu nhiều chiều được giảm xuống 2 hoặc 3 chiều, chúng ta có thể dễ dàng vẽ biểu đồ và phát hiện xu hướng.

Với những lợi ích này, PCA là một công cụ quan trọng trong phân tích dữ liệu, đặc biệt trong các lĩnh vực như **thị giác máy tính, xử lý ảnh, tài chính và sinh học**. 🚀

1. Problem Statement

**CHƯƠNG 2. PHÁT BIỂU BÀI TOÁN**

**2.1. Mô tả bài toán**

Phân tích thành phần chính (Principal Component Analysis - PCA) là một phương pháp giảm chiều dữ liệu, giúp loại bỏ sự dư thừa thông tin và giữ lại những đặc trưng quan trọng nhất của dữ liệu. Bài toán được đặt ra trong bối cảnh dữ liệu có số chiều lớn, gây khó khăn trong việc tính toán, trực quan hóa và phân tích. PCA giúp tìm ra một tập hợp các trục tọa độ mới sao cho dữ liệu được biểu diễn tối ưu trong một không gian có số chiều thấp hơn.

**Vấn đề gặp phải khi dữ liệu có số chiều cao:**

* Tốn nhiều tài nguyên tính toán.
* Dữ liệu có thể bị nhiễu, chứa nhiều thông tin dư thừa.
* Khó trực quan hóa khi số chiều lớn hơn 3.
* Hiệu suất của các mô hình học máy có thể bị giảm do vấn đề 'lời nguyền số chiều' (Curse of Dimensionality).

Do đó, cần một phương pháp có thể giảm số chiều dữ liệu mà vẫn bảo toàn được phần lớn thông tin quan trọng. PCA là một trong những phương pháp phổ biến để giải quyết vấn đề này.

**2.2. Input và Output của hệ thống**

* **Input**: Một tập dữ liệu XX có dạng:

X={x1,x2,...,xn}∈Rm×nX = \{x\_1, x\_2, ..., x\_n\} \in \mathbb{R}^{m \times n}

Trong đó:

* + mm là số lượng mẫu dữ liệu.
  + nn là số chiều của mỗi mẫu dữ liệu.
* **Output**: Một tập dữ liệu mới X′X' có số chiều thấp hơn nhưng vẫn giữ lại phần lớn thông tin ban đầu:

X′={x1′,x2′,...,xm′}∈Rm×k,k<nX' = \{x'\_1, x'\_2, ..., x'\_m\} \in \mathbb{R}^{m \times k}, k < n

Trong đó kk là số chiều mới được chọn sao cho phần lớn phương sai của dữ liệu được giữ lại.

**2.3. Framework chung của hệ thống**

Hệ thống thực hiện PCA có thể được chia thành các công đoạn chính sau:

1. **Tiền xử lý dữ liệu**:
   * Chuẩn hóa dữ liệu về cùng một phân phối (thường là trung bình 0, phương sai 1).
   * Loại bỏ dữ liệu ngoại lai nếu cần thiết.
2. **Tính toán ma trận hiệp phương sai**:
   * Xác định mối quan hệ tuyến tính giữa các biến trong tập dữ liệu bằng cách tính toán ma trận hiệp phương sai.
3. **Tính toán vector riêng và giá trị riêng**:
   * Tìm các thành phần chính bằng cách giải bài toán trị riêng của ma trận hiệp phương sai.
4. **Chọn số lượng thành phần chính**:
   * Lựa chọn số chiều kk dựa trên tỷ lệ phương sai tích lũy.
5. **Biểu diễn dữ liệu trong không gian mới**:
   * Chiếu dữ liệu gốc vào không gian mới với số chiều giảm.

**CHƯƠNG 3. PHƯƠNG PHÁP**

Trong chương này, chúng ta sẽ trình bày một phương pháp tiên tiến để thực hiện PCA một cách hiệu quả. Một trong những cách tiếp cận phổ biến và hiệu quả nhất là **PCA sử dụng Phân rã Giá trị Kỳ dị (Singular Value Decomposition - SVD)**.

**3.1. Các bước thực hiện PCA sử dụng SVD**

**Bước 1: Chuẩn hóa dữ liệu**

Trước khi thực hiện PCA, cần chuẩn hóa dữ liệu để đảm bảo các biến có cùng đơn vị đo lường và ảnh hưởng tương đương:

Xnormalized=X−μσX\_{normalized} = \frac{X - \mu}{\sigma}

Trong đó:

* μ\mu là vector trung bình của từng đặc trưng.
* σ\sigma là độ lệch chuẩn của từng đặc trưng.

**Bước 2: Tính toán ma trận hiệp phương sai**

Ma trận hiệp phương sai được tính bằng công thức:

C=1mXTXC = \frac{1}{m} X^T X

Trong đó:

* CC là ma trận hiệp phương sai kích thước n×nn \times n.
* XX là dữ liệu đã được chuẩn hóa.

**Bước 3: Phân rã giá trị kỳ dị (SVD)**

Thay vì tính toán vector riêng của ma trận hiệp phương sai, có thể sử dụng phương pháp SVD:

X=USVTX = U S V^T

Trong đó:

* UU là ma trận trực giao chứa các vector riêng của XXTXX^T.
* SS là ma trận đường chéo chứa các giá trị riêng.
* VV là ma trận trực giao chứa các vector riêng của XTXX^T X.

Các thành phần chính chính là các hàng của ma trận VTV^T.

**Bước 4: Chọn số thành phần chính**

Số chiều mới kk được chọn dựa trên phương sai tích lũy. Tổng phương sai được tính bằng:

Explained Variance=∑i=1ksi∑i=1nsi\text{Explained Variance} = \frac{\sum\_{i=1}^{k} s\_i}{\sum\_{i=1}^{n} s\_i}

Trong đó sis\_i là các giá trị trên đường chéo của SS.

Số kk được chọn sao cho phương sai tích lũy đạt một ngưỡng nhất định (ví dụ: 95%).

**Bước 5: Chiếu dữ liệu vào không gian mới**

Sau khi chọn kk, dữ liệu được chuyển đổi bằng cách nhân với VkV\_k (chứa kk thành phần chính đầu tiên của VV):

X′=XVkTX' = X V\_k^T

Dữ liệu mới X′X' có số chiều kk, giúp giảm tải tính toán và giữ được phần lớn thông tin.

**3.2. So sánh với các phương pháp khác**

* **PCA sử dụng Eigen Decomposition**: Dùng phương pháp trị riêng để tính toán PCA, nhưng kém hiệu quả hơn SVD khi dữ liệu lớn.
* **Autoencoder (Deep Learning)**: Một phương pháp phi tuyến tính có thể học biểu diễn dữ liệu tốt hơn PCA, nhưng yêu cầu nhiều tài nguyên tính toán.
* **t-SNE & UMAP**: Các phương pháp giảm chiều hiện đại giúp trực quan hóa dữ liệu tốt hơn PCA, nhưng không bảo toàn cấu trúc tuyến tính.

**3.3. Ứng dụng thực tế**

* **Xử lý ảnh**: Giảm số chiều của tập dữ liệu ảnh để tăng tốc huấn luyện mô hình học sâu.
* **Phân tích tài chính**: PCA giúp xác định các yếu tố chính ảnh hưởng đến giá cổ phiếu.
* **Nhận dạng khuôn mặt**: PCA trích xuất các đặc trưng quan trọng từ ảnh khuôn mặt, giúp tăng độ chính xác của các thuật toán nhận diện.

Với phương pháp SVD, PCA có thể thực hiện hiệu quả trên tập dữ liệu lớn, giúp giảm chiều nhanh chóng mà vẫn giữ được thông tin quan trọng.

1. Method
2. Geometrical Explanation
3. Model Checking
4. Case Study