Handsign Recognition

Using LaTeX to prepare slides

Nguyen Vi Chi Cuong Do Tien Dung Do Huu Dat
VNU Hanoi University of Science
Ngày 20 tháng 4 năm 2025

Mục lục

Giới thiệu Bài toán: Nhận diện Bảng chữ cái ASL qua Hình

Ngôn ngữ Ký hiệu Mỹ (ASL):

Là ngôn ngữ giao tiếp chính của cộng đồng người Điếc ở Mỹ và Canada.

Bao gồm các ký hiệu dùng tay, cử động cơ thể và biểu cảm khuôn mặt.

Bảng chữ cái ngón tay (Finger Spelling):

Một phần của ASL, dùng để đánh vần các từ (thường là tên riêng, địa điểm, từ mượn).

Mỗi chữ cái trong bảng chữ cái Latin (A-Z) được biểu diễn bằng một hình dạng bàn tay tĩnh.

Bài toán đặt ra:

Xây dựng hệ thống tự động nhận diện chính xác chữ cái ASL từ hình ảnh tĩnh của bàn tay.

Mục tiêu là chuyển đổi hình ảnh thành văn bản chữ cái tương ứng.

Tại sao cần Học máy?

Hình ảnh chứa thông tin phức tạp (hình dạng bàn tay, góc chụp, ánh sáng, nền).

Các mô hình học máy có khả năng tự động học và trích xuất các đặc trưng hữu ích từ dữ liệu hình ảnh thô.

Bài toán Nhận diện ASL: Ý nghĩa và Thách thức

Ý nghĩa:

Tăng cường khả năng giao tiếp và tiếp cận thông tin cho cộng đồng người Điếc.
Phát triển các ứng dụng dịch thuật ngôn ngữ ký hiệu thời gian thực.
Hỗ trợ giáo dục và đào tạo ngôn ngữ ký hiệu.
Cải thiện tương tác giữa con người và máy tính (HCI).

Thách thức chính:

Biến thể hình ảnh: Sự khác biệt về hình dạng bàn tay, kích thước, màu da giữa các cá nhân.
Điều kiện môi trường: Ảnh hưởng của ánh sáng, bóng đổ, nền ảnh phức tạp.
Góc nhìn: Sự thay đổi khi chụp từ các góc khác nhau.
Tương đồng ký hiệu: Một số chữ cái ASL có hình dạng rất giống nhau (ví dụ: E, M, N, S, T).
Thiếu dữ liệu: Việc thu thập tập dữ liệu lớn và đa dạng có thể khó khăn.

Mục tiêu: Tìm hiểu cách các mô hình học máy (như Softmax Regression, CNNs,...) được sử dụng để giải quyết bài toán này và đánh giá khả năng của chúng.

Mục lục

Tiền xử lý dữ liệu

Mục tiêu: Chuẩn bị dữ liệu ảnh để đồng nhất đầu vào cho mô hình học máy.

Các bước chính:

Resize ảnh: Chuẩn hóa kích thước ảnh.
Chuẩn hóa dữ liệu: Đưa dữ liệu về cùng phạm vi giá trị.
Phân tích thành phần chính (PCA): Giảm số chiều dữ liệu.

Ứng dụng: Tăng hiệu quả cho mô hình, ví dụ: Gaussian Naive Bayes.

Resize ảnh

Mục tiêu: Chuẩn hóa kích thước ảnh từ 64×64 về 28×28.

Phương pháp: Nội suy tuyến tính kép (Bilinear Interpolation).

Cách hoạt động:

Lấy 4 pixel lân cận (Q11, Q21, Q12, Q22) để tính giá trị pixel mới.
Công thức:

$$\text{Giá trị mới} = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{11} + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{21} + \dots$$

Ví dụ: Tại (10, 10) trong ảnh mới, tọa độ ánh xạ (22.857, 22.857), giá trị pixel: 115.71.

Tác động: Giảm số chiều từ 4096 (64×64) xuống 784 (28×28), giữ chi tiết hình dạng tay.

Chuẩn hóa dữ liệu

Mục tiêu: Đưa dữ liệu về cùng phạm vi, tránh đặc trưng lớn lấn át mô hình.

Phương pháp: Z-score normalization (trung bình = 0, độ lệch chuẩn = 1).

Công thức:

$$\text{Giá trị mới} = \frac{\text{Giá trị} - \text{Trung bình}}{\text{Độ lệch chuẩn}}$$

Ví dụ: Pixel 15.9, trung bình 10, độ lệch chuẩn 4:

$$\frac{15.9 - 10}{4} = 1.475$$

Tác động: Đảm bảo các đặc trưng (pixel) có cùng mức ảnh hưởng, hỗ trợ Gaussian Naive Bayes.

Phân tích thành phần chính (PCA)

Mục tiêu: Giảm số chiều, loại bỏ nhiễu, giảm chi phí tính toán.

Phương pháp: Chiếu dữ liệu lên các thành phần chính có phương sai lớn nhất.

Các bước:

Chuẩn hóa dữ liệu.
Tính ma trận hiệp phương sai.
Tìm giá trị riêng, vector riêng, chọn k thành phần chính.
Chiếu dữ liệu: Dữ liệu mới = Dữ liệu gốc $\times$ Ma trận thành phần chính.

Ví dụ: Từ 784 chiều (28×28) xuống 50 chiều, giữ 95% phương sai.

Tác động: Giảm nhiễu, giữ thông tin chính về hình dạng tay trong ASL.

Mục lục

Gaussian Naive Bayes

Gaussian Naive Bayes: Một biến thể của Naive Bayes Classifier.

Ứng dụng:

Phân loại dữ liệu liên tục (giá trị pixel, cảm biến).
Dữ liệu số chiều cao sau tiền xử lý.

Đặc điểm chính:

Dựa trên Định lý Bayes.
Giả định đặc trưng độc lập và tuân theo phân phối chuẩn.

Nguyên lý và công thức của Gaussian Naive Bayes

Nguyên lý:

Dựa trên **Định lý Bayes**:

$$P(c|x) = \frac{P(c) \cdot P(x|c)}{P(x)}$$

Giả định độc lập: Các đặc trưng (pixel) độc lập với nhau.

Giả định phân phối chuẩn: Dữ liệu tuân theo phân phối chuẩn cho từng lớp (A đến Z).

Công thức:

Xác suất lớp c cho mẫu x:

$$P(c|x) \propto P(c) \cdot P(x|c)$$

Xác suất $P(x_i|c)$:

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_{c,i}^2}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

Ưu điểm và hạn chế của Gaussian Naive Bayes

Ưu điểm:

Huấn luyện nhanh, phù hợp với dữ liệu liên tục (giá trị pixel).
Hiệu quả với **số chiều lớn** sau tiền xử lý (ví dụ: 50 chiều).

Hạn chế:

Giả định **độc lập** giữa các đặc trưng không đúng với dữ liệu ảnh (pixel có tương quan).
Hiệu suất thấp nếu dữ liệu **không tuân theo phân phối chuẩn**.

Softmax Regression là gì?

Là một thuật toán phân loại tuyến tính (linear classification algorithm).

Là sự mở rộng của Logistic Regression cho bài toán có **nhiều hơn hai lớp** (multi-class classification).

Thích hợp cho các bài toán mà **mỗi mẫu chỉ thuộc về DUY NHẤT một lớp**.

Đầu ra là phân phối xác suất cho từng lớp, cho biết khả năng mẫu thuộc về lớp nào.

Nguyên lý Hoạt động: Tính điểm (Scores)

Mỗi hình ảnh đầu vào (sau khi xử lý) được biểu diễn dưới dạng một vector đặc trưng x.
Mô hình học một bộ trọng số W (ma trận) và độ lệch b (vector).
Đối với mỗi lớp k (trong bài toán ASL là 26 lớp A-Z), mô hình tính một điểm (score) hoặc logit z_k bằng cách kết hợp tuyến tính:

$$z_k = w_k^T x + b_k$$

Dạng ma trận:

$$Z = Wx + b$$

Trong đó:

$x \in \mathbb{R}^d$ là vector đặc trưng đầu vào (ví dụ: pixel của ảnh).
$W \in \mathbb{R}^{K \times d}$ là ma trận trọng số.
$b \in \mathbb{R}^K$ là vector độ lệch.
$Z \in \mathbb{R}^K$ là vector điểm cho từng lớp.

Các điểm này có thể là bất kỳ giá trị thực nào (dương, âm hoặc không).

Nguyên lý Hoạt động: Hàm Softmax

Để biến các điểm z_k thành xác suất, chúng ta sử dụng hàm Softmax.

Hàm Softmax chuẩn hóa các điểm sao cho chúng nằm trong khoảng $(0, 1)$ và tổng bằng 1.

Công thức:

$$P(y = k \mid x; W, b) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

Trong đó:

e^{z_k} là lũy thừa của điểm z_k (luôn dương).
$\sum_{j=1}^{K} e^{z_j}$ là tổng chuẩn hóa.

Vector xác suất: $[\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K]$, với $\sum \hat{y}_k = 1$.

Dự đoán là lớp có xác suất cao nhất:

$$\hat{y} = \arg\max_k P(y = k \mid x; W, b)$$

Nguyên lý Hoạt động: Hàm mất mát và Huấn luyện

Hàm mất mát dùng để đo sự khác biệt giữa dự đoán và nhãn thực tế.
Dùng hàm **Cross-Entropy Loss**:

$$L(\hat{y}, y) = -\sum_{k=1}^{K} y_k \log(\hat{y}_k)$$

Mục tiêu: giảm trung bình loss trên toàn bộ tập dữ liệu:

$$J(W, b) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(\hat{y}_{ik})$$

Dùng Gradient Descent để cập nhật:

$$W \leftarrow W - \alpha \nabla_W L, \quad b \leftarrow b - \alpha \nabla_b L$$

Công thức Tổng quát

Đầu vào: $x \in \mathbb{R}^d$

Trọng số và độ lệch: $W \in \mathbb{R}^{K \times d}$, $b \in \mathbb{R}^K$

Logits: $Z = Wx + b$

Softmax:

$$P(y = k \mid x; W, b) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

Cross-Entropy Loss:

$$L(W, b; x, y) = -\sum_{k=1}^{K} y_k \log \left(\frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \right)$$

Cập nhật:

$$W \leftarrow W - \alpha \nabla_W L, \quad b \leftarrow b - \alpha \nabla_b L$$

Áp dụng Softmax Regression cho Nhận diện ASL

Dữ liệu: hình ảnh bàn tay biểu diễn chữ cái ASL.

Tiền xử lý:

Resize ảnh (VD: 28x28).
Chuyển sang ảnh xám.
Chuẩn hóa pixel về [0, 1].
Làm phẳng thành vector $x \in \mathbb{R}^{784}$.

Số lớp: $K = 26$

Mô hình:

Đầu vào: $d = 784$, đầu ra: $K = 26$
Kích hoạt Softmax ở đầu ra.

Huấn luyện: sử dụng Cross-Entropy và Gradient Descent

Dự đoán: lấy lớp có xác suất cao nhất

Ưu điểm của Softmax Regression

Đơn giản và dễ hiểu:

Nguyên lý trực quan, dễ triển khai.
Phù hợp làm baseline.

Tính toán hiệu quả:

Nhanh, nhẹ, phù hợp cho hệ thống thời gian thực.

Đầu ra xác suất:

Biểu diễn mức độ tự tin của mô hình.
Kết hợp tốt với hệ thống phân loại phức hợp.

Hạn chế của Softmax Regression

Tuyến tính: không học được các đặc trưng phi tuyến.

Phụ thuộc đặc trưng: hiệu suất giảm nếu đặc trưng không đủ mạnh.

Mất cấu trúc không gian ảnh: khi làm phẳng ảnh.

Nhạy cảm với nhiễu: dễ bị ảnh hưởng nếu không chuẩn hóa tốt.

K-Nearest Neighbors (KNN)

KNN: Một thuật toán học có giám sát.

Ứng dụng:

Phân loại (classification) và hồi quy (regression).
Nhận diện mẫu, phân loại văn bản, hệ thống gợi ý.

Đặc điểm chính:

Thuộc nhóm phương pháp *lazy learning* (học lười).
Dựa trên khoảng cách giữa các điểm dữ liệu.
Đơn giản, dễ triển khai, không yêu cầu giả định về phân phối dữ liệu.

Nguyên lý và công thức của KNN

Nguyên lý:

Dự đoán nhãn của điểm dữ liệu mới dựa trên nhãn của k láng giềng gần nhất.
Tính khoảng cách giữa điểm mới x và các điểm trong tập huấn luyện.
Chọn k điểm gần nhất và áp dụng quy tắc đa số hoặc có trọng số.

Công thức khoảng cách:

Khoảng cách Euclid:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{p} (x_j - x_{ij})^2}$$

Khoảng cách Minkowski:

$$d(x, x_i) = \left(\sum_{j=1}^{p} |x_j - x_{ij}|^q \right)^{1/q}$$

Quy tắc phân loại và dự đoán

Chọn k láng giềng gần nhất:

Sắp xếp các điểm huấn luyện theo khoảng cách tăng dần đến x.

Lấy k điểm gần nhất tạo thành tập $N_k(x)$.

Quy tắc phân loại:

Đa số phiếu bầu:

$$y = \arg\max_{C_j} \sum_{(x_i, y_i) \in N_k(x)} I(y_i = C_j)$$

Phiếu bầu có trọng số:

$$y = \arg\max_{C_j} \sum_{(x_i, y_i) \in N_k(x)} w_i \cdot I(y_i = C_j)$$

với $w_i = \frac{1}{d(x, x_i)}$.

Dự đoán xác suất:

$$P(C_j) = \frac{N_j}{k}$$

Các tham số quan trọng

Giá trị k:

k nhỏ: Mô hình nhạy cảm với nhiễu, dễ bị *overfitting*.
k lớn: Mô hình tổng quát hơn, nhưng có thể bỏ sót chi tiết, dẫn đến *underfitting*.
Lựa chọn k tối ưu: Dựa trên kiểm định chéo (*cross-validation*).

Thước đo khoảng cách:

Euclid, Manhattan, Minkowski, v.v.
Lựa chọn dựa trên đặc điểm dữ liệu.

Chuẩn hóa dữ liệu:

Cần thiết để đảm bảo các đặc trưng đóng góp công bằng.
Tránh đặc trưng có độ lớn lớn chi phối khoảng cách.

Mục lục

Kết quả các chỉ số của ba mô hình

Mục tiêu: So sánh ba mô hình học máy trên bài toán nhận diện bảng chữ cái ASL dựa trên các chỉ số:

Accuracy, *Precision*, *Recall*, *Confusion Matrix*

Dữ liệu đầu vào: Vector pixel từ ảnh ASL đã qua resize, chuẩn hóa và PCA.

Chỉ số hiệu suất: GNB và MLR

Gaussian Naive Bayes (GNB):

- Train time: 0.0316 giây
- Accuracy: 0.5939
- Precision: 0.6616
- Recall: 0.5939

Multinomial Logistic Regression (MLR):

- Train time: 1.4367 giây
- Accuracy: 0.7680
- Precision: 0.7706
- Recall: 0.7680

Chỉ số hiệu suất: KNN và bảng tổng hợp

K-Nearest Neighbors (KNN):

- Train time: 0.0147 giây
- Accuracy: 0.9928
- Precision: 0.9928
- Recall: 0.9928

Bảng tổng hợp:

Mô hình	Train Time (s)	Accuracy	Precision	Recall
GNB	0.0316	0.5939	0.6616	0.5939
MLR	1.4367	0.7680	0.7706	0.7680
KNN	0.0147	0.9928	0.9928	0.9928

Phân tích kết quả các mô hình

- **Thời gian huấn luyện:** KNN nhanh nhất (0.0147s), tiếp theo là GNB, chậm nhất là MLR.
- **Độ chính xác:** KNN cao nhất (99.28%), vượt trội MLR (76.80%) và GNB (59.39%).
- **Precision và Recall:** KNN rất tốt và cân bằng, MLR khá tốt, GNB thấp do giả định đơn giản.