Ứng dụng SVM phi tuyến kết hợp với PCA cho bài toán phân loại hình ảnh

Nguyen Viet Phuong
MSSV: *24022431*

Ngày 14 tháng 1 năm 2026

Mục lục

Chương 1

Giới thiệu về chủ đề

1.1 Bối cảnh và động cơ nghiên cứu

Trong những năm gần đây, bài toán phân loại hình ảnh (Image Classification) đã trở thành một trong những bài toán trung tâm của lĩnh vực Thị giác máy tính (Computer Vision) và Trí tuệ nhân tạo (Artificial Intelligence). Với sự phát triển mạnh mẽ của các mô hình học máy và học sâu, việc tự động nhận diện và phân loại đối tượng trong bức ảnh được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, công nghiệp hay nông nghiệp.

Tuy nhiên, không phải lúc nào ta cũng có điều kiện về chi phí và dữ liệu để có thể triển khai triệt để công việc này. Trong nhiều trường hợp, đặc biệt là các bài toán có quy mô dữ liệu vừa phải, việc kết hợp các phương pháp trích xuất đặc trưng truyền thống với các mô hình học máy cổ điển vẫn mang lại hiệu quả cao, đồng thời tăng khả năng diễn giải và giảm thiểu chi phí huấn luyện.

Xuất phát từ những điều này, bài báo cáo tập trung nghiên cứu bài toán phân loại ảnh nhị phân giữa *Shells* và *Pebbles*, dựa trên việc kết hợp hai phương pháp quan trọng trong Machine Learning: Máy vector hỗ trợ phi tuyến (Non-Linear Support Vector Machine) và Phân tích thành phần chính (Principal Component Analysis).

1.2 Mô tả bài toán

Bài toán đặt ra yêu cầu xây dựng một hệ thống học máy có khả năng tự động phân loại ảnh đầu vào thành một trong hai lớp:

- **Shells**: Ảnh chứa các vỏ sinh vật biển với hình dạng và hoa văn phức tạp
- **Pebbles**: Ảnh chứa các viên sỏi, đá nhỏ với bề mặt tương đối trơn và cấu trúc ngẫu nhiên.

Tập dữ liệu gồm 4284 bức ảnh, với 2743 ảnh Pebbles và 1541 ảnh Shells. Các ảnh có kích thước và tỉ lệ khác nhau, ánh sáng cũng như bối cảnh chụp đa dạng, làm tăng độ phức tạp trong quá trình xử lí hình ảnh.

Chương 2

Cơ sở lý thuyết

2.1 Support Vector Machine (SVM)

2.1.1 Bài toán phân loại nhị phân

Xét tập dữ liệu huấn luyện:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, +1\}.$$

Mục tiêu của Support Vector Machine là tìm một hàm quyết định:

$$f(x) = w^T x - b$$

sao cho các mẫu dữ liệu thuộc hai lớp được phân tách với *khoảng cách biên* (margin) lớn nhất.

2.1.2 Linear SVM và bài toán tối ưu

Khoảng cách hình học từ một điểm x_i đến siêu phẳng $w^T x - b = 0$ được cho bởi:

$$\frac{|w^T x_i - b|}{\|w\|}.$$

Để tối đa hóa khoảng cách biên giữa hai lớp, Linear SVM giải bài toán tối ưu lồi:

$$\min_{w,b} \ \frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(w^T x_i - b) \geq 1, \ \forall i.$$

Bài toán trên được gọi là **hard-margin SVM**, giả định rằng dữ liệu phân tách tuyến tính hoàn hảo.

2.1.3 Soft-margin SVM và hàm mất mát Hinge

Trong thực tế, dữ liệu thường có nhiễu và không thể phân tách hoàn hảo. Khi đó, ta đưa vào các biến trượt $\xi_i \geq 0$:

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{s.t.} \quad y_i(w^T x_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

Bài toán này tương đương với việc tối ưu hàm mất mát:

$$\mathcal{L}(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \max\big(0, 1 - y_i f(x_i)\big),$$

trong đó

$$\ell_{\text{hinge}}(y, f(x)) = \max(0, 1 - yf(x))$$

được gọi là **hinge loss**.

2.1.4 Non-Linear SVM và Kernel Trick

Khi dữ liệu không thể phân tách tuyến tính trong không gian ban đầu, SVM sử dụng một ánh xạ phi tuyến:

$$\phi : \mathbb{R}^d \to \mathcal{H},$$

trong đó $\mathcal{H}$ là không gian đặc trưng có số chiều cao hơn.

Hàm quyết định khi đó có dạng:

$$f(x) = w^T \phi(x) - b.$$

Bài toán tối ưu trở thành:

$$\min_w \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \max\big(0, 1 - y_i(w^T\phi(x_i) - b)\big).$$

Thay vì tính $\phi(x)$ một cách tường minh, SVM sử dụng **kernel trick**:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle.$$

Một số kernel phổ biến:

$$\begin{aligned} \text{Linear: } & K(x, z) = x^T z, \\ \text{Polynomial: } & K(x, z) = (x^T z + c)^p, \\ \text{RBF: } & K(x, z) = \exp(-\gamma \|x - z\|^2). \end{aligned}$$

Trong báo cáo này, ánh xạ đa thức bậc hai:

$$\phi(x) = [x, x^2]$$

được sử dụng để mô hình hóa ranh giới quyết định phi tuyến.

2.2 Principal Component Analysis (PCA)

2.2.1 Mục tiêu của PCA

Cho dữ liệu đã được chuẩn hóa:

$$X = [x_1, x_2, \ldots, x_N]^T \in \mathbb{R}^{N \times d}.$$

PCA tìm một hướng chiếu w sao cho phương sai của dữ liệu sau khi chiếu lên w là lớn nhất:

$$\max_w \ \text{Var}(Xw) \quad \text{s.t.} \quad \|w\| = 1.$$

2.2.2 Dạng toán học

Phương sai của dữ liệu sau chiếu được viết dưới dạng:

$$\mathrm{Var}(Xw) = w^T \Sigma w,$$

trong đó ma trận hiệp phương sai:

$$\Sigma = \frac{1}{N} X^T X.$$

Bài toán PCA tương đương với:

$$\max_{w} \; w^T \Sigma w \quad \text{s.t.} \quad w^T w = 1.$$

2.2.3 Chứng minh bằng nhân tử Lagrange

Xét hàm Lagrange:

$$\mathcal{L}(w, \lambda) = w^T \Sigma w - \lambda (w^T w - 1).$$

Lấy đạo hàm theo w:

$$\frac{\partial \mathcal{L}}{\partial w} = 2\Sigma w - 2\lambda w = 0.$$

Suy ra:

$$\Sigma w = \lambda w.$$

Do đó, các thành phần chính là các vector riêng của Σ ứng với các trị riêng lớn nhất.

2.2.4 Giảm chiều dữ liệu

Chọn k vector riêng ứng với k trị riêng lớn nhất:

$$W_k = [w_1, w_2, \ldots, w_k].$$

Dữ liệu sau khi giảm chiều:

$$Z = XW_k.$$

Tỉ lệ phương sai được giữ lại:

$$\mathrm{EVR}(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{d} \lambda_j}.$$

2.3 Vai trò của PCA trong SVM

Sau PCA, bài toán SVM được giải trong không gian $\mathbb{R}^k$:

$$\min_{u,b} \frac{1}{2}\|u\|^2 + C\sum_{i=1}^{N} \max\big(0, 1 - y_i(u^T z_i - b)\big),$$

trong đó $u \in \mathbb{R}^k$.

Vector trọng số trong không gian ban đầu có dạng:

$$w = W_k u.$$

Do W_k là ma trận trực chuẩn, ta có:

$$\|w\|^2 = \|W_k u\|^2 = u^T W_k^T W_k u = \|u\|^2.$$

Điều này cho thấy PCA không làm thay đổi bản chất điều chuẩn của SVM, nhưng loại bỏ các thành phần của w nằm trong không gian con ứng với các trị riêng nhỏ, tức các hướng có phương sai thấp của dữ liệu.

Mặc dù PCA là phép biến đổi tuyến tính, việc áp dụng PCA trước Non-Linear SVM không làm mất khả năng học ranh giới phi tuyến. Sau PCA, ánh xạ phi tuyến được áp dụng trên dữ liệu đã giảm chiều:

$$f(x) = w^T \phi(W_k^T x) - b.$$

Do đó, PCA đóng vai trò giảm chiều và điều hòa không gian đặc trưng, trong khi Non-Linear SVM chịu trách nhiệm mô hình hóa ranh giới quyết định phi tuyến.

Chương 3

Phương pháp huấn luyện

3.1 Kiến trúc tổng thể của mô hình

Mô hình phân loại ảnh này được xây dựng theo kiến trúc pipeline học máy truyền thống, bao gồm cách khối xử lý tuần tự từ dữ liệu ảnh thô đến bộ phân loại cuối cùng. Không giống các mô hình học sâu học đặc trưng trực tiếp từ ảnh, kiến trúc đề xuất tách biệt rõ ràng giữa giai đoạn trích xuất đặc trưng và giai đoạn học mô hình.

Cụ thể, kiến trúc tổng thể của mô hình gồm các thành phần chính sau:

1. Đọc và kiểm tra dữ liệu ảnh từ tập dữ liệu gốc
2. Tiền xử lý và chuẩn hóa ảnh đầu vào
3. Biểu diễn ảnh dưới dạng vector đặc trưng có số chiều cao
4. Chuẩn hóa dữ liệu đặc trưng (mean = 0, std = 1)
5. Giảm chiều dữ liệu bằng Principal Component Analysis (PCA)
6. Huấn luyện mô hình Non-Linear Support Vector Machine
7. Đánh giá mô hình trên tập kiểm thử

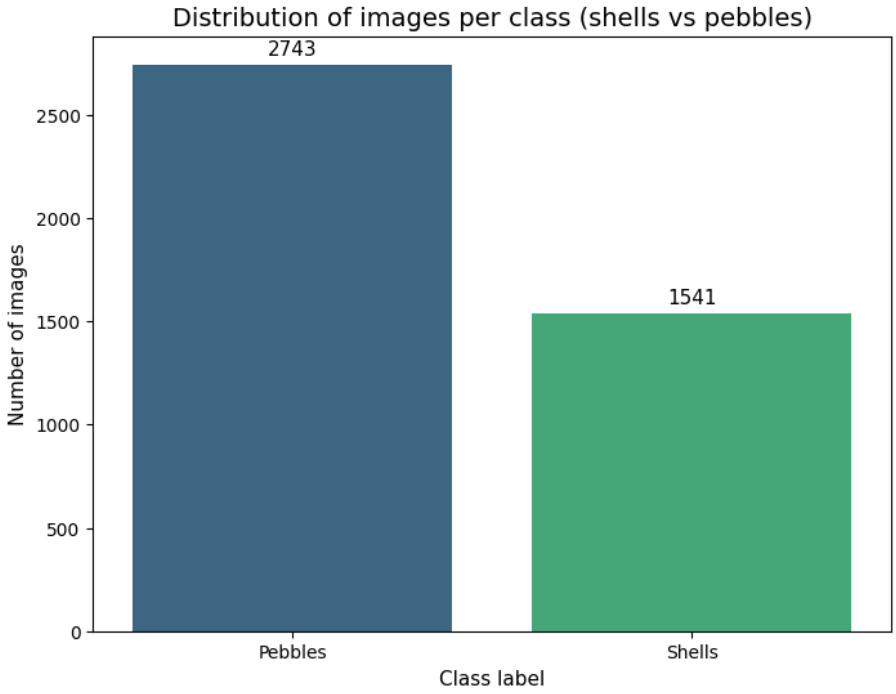
Cách tiếp cận này cho phép kiểm soát chặt chẽ từng bước xử lý, đồng thời đảm bảo sự nhất quán giữa mô tả lý thuyết và thực nghiệm.

3.2 Tập dữ liệu và nguồn dữ liệu

Tập dữ liệu được sử dụng trong nghiên cứu là *Shells or Pebbles Dataset* được công bố trên nền tảng Kaggle.

Các đặc điểm chính của tập dữ liệu:

- Tổng số ảnh: 4284
- Số lớp: 2 (Shells và Pebbles)
- Số ảnh Pebbles: 2743
- Số ảnh Shells: 1541



Hình 3.1: Thống kê tập dữ liệu ảnh Shells & Pebbles

Hình 3.1 cho thấy sự mất cân bằng tương đối giữa hai lớp dữ liệu, trong đó số lượng ảnh thuộc lớp Pebbles lớn hơn đáng kể so với lớp Shells. Đặc điểm này có thể ảnh hưởng đến khả năng tổng quát hóa của mô hình và được xem xét trong quá trình đánh giá kết quả.

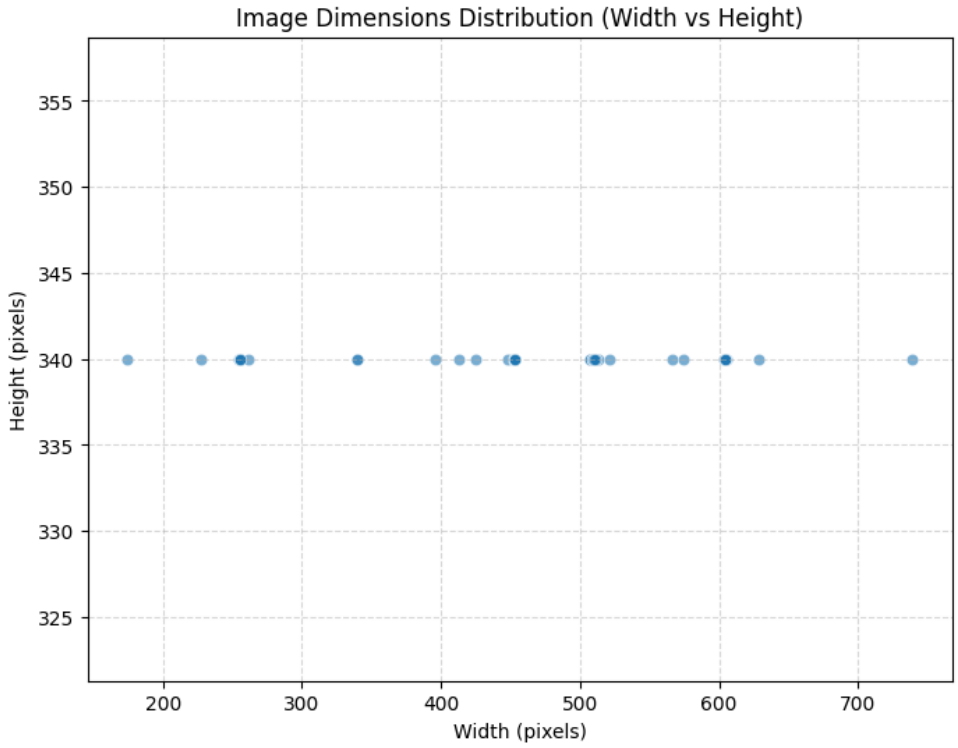
Hình 3.2: Năm ảnh ngẫu nhiên được trích xuất từ mỗi lớp dữ liệu

Hình 3.2 minh họa sự đa dạng về hình dạng, màu sắc và kết cấu trong mỗi lớp dữ liệu. Đặc biệt, các ảnh thuộc lớp Shells thể hiện sự phức tạp cao về cấu trúc hình học, trong khi lớp Pebbles có xu hướng đồng nhất hơn về mặt hình dạng và texture.

3.3 Tiền xử lý dữ liệu ảnh

Các ảnh trong tập dữ liệu ban đầu có sự khác biệt đáng kể về kích thước, tỷ lệ khung hình và định dạng lưu trữ. Do đó, một chuỗi các bước tiền xử lý được thiết kế

nhằm đảm bảo tính đồng nhất của dữ liệu đầu vào trước khi đưa vào mô hình SVM.



Hình 3.3: Hình 3. Phân bố kích thước ảnh trong tập dữ liệu (Width vs Height)

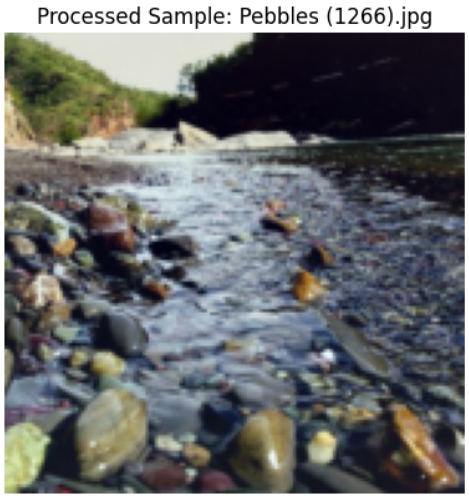
Hình 3.3 cho thấy chiều cao của các ảnh trong Dataset rất ổn định, tuy nhiên chiều rộng có sự dao động đáng kể. Điều này là cản trở không nhỏ nếu ta thực hiện đưa trực tiếp ảnh thô vào mô hình, bởi sự không đồng nhất về tỷ lệ ảnh giữa các mẫu dữ liệu. Vì vậy, các bước tiền xử lý nhằm chuẩn hóa kích thước và tỷ lệ ảnh là điều bắt buộc.

Quy trình tiền xử lý ảnh được thực hiện theo các bước sau:

- Các ảnh grayscale được chuyển sang ảnh RGB bằng cách nhân bản kênh xám thành ba kênh màu.
- Các ảnh có kênh alpha (RGBA) được loại bỏ kênh alpha, chỉ giữ lại ba kênh màu RGB.
- Thực hiện phép *center-crop* để cắt ảnh về dạng hình vuông, đảm bảo giữ lại vùng trung tâm mang nhiều thông tin nhất.
- Ảnh sau khi cắt được resize về kích thước cố định 150×150 bằng nội suy song tuyến (bilinear interpolation).
- Giá trị pixel được chuẩn hóa về miền $[0, 1]$ nhằm ổn định quá trình huấn luyện mô hình.

Sau quy trình trên, mỗi ảnh được biểu diễn dưới dạng tensor có kích thước $150 \times 150 \times 3$, ở đây tensor ảnh được "duỗi phẳng"để thành vector đặc trưng có số chiều:

$$d = 150 \times 150 \times 3 = 67\,500.$$



Hình 3.4: Ảnh ví dụ sau quy trình xử lý

3.4 Chuẩn hóa và giảm chiều dữ liệu

Sau quy trình tiền xử lý và duỗi phẳng ảnh, vấn đề ở đây là mỗi ảnh được biểu diễn trong không gian đặc trưng với số chiều rất lớn:

$$d = 150 \times 150 \times 3 = 67\,500.$$

Không gian đặc trưng chiều cao này đặt ra nhiều thách thức, khi số chiều đặc trưng lớn hơn rất nhiều so với số lượng mẫu huấn luyện, dữ liệu rất dễ rơi vào hiện tượng *Curse of Dimensionality*, hay *Lời nguyền của số chiều*, khiến khoảng cách giữa các điểm dữ liệu trở nên kém phân biệt và làm tăng xác suất overfitting.

3.4.1 Chuẩn hóa dữ liệu đặc trưng

Trước khi áp dụng PCA, dữ liệu được chuẩn hóa thủ công theo từng chiều đặc trưng dựa trên tập huấn luyện. Cụ thể, với mỗi đặc trưng x, phép chuẩn hóa được thực hiện theo công thức:

$$x' = \frac{x - \mu}{\sigma},$$

trong đó μ và σ lần lượt là giá trị trung bình và độ lệch chuẩn của đặc trưng, được tính trên tập huấn luyện. Phép chuẩn hóa này giúp đảm bảo rằng các đặc trưng có cùng thang đo, đồng thời tránh việc một số chiều có phương sai lớn chi phối quá trình học của mô hình.

3.4.2 Giảm chiều bằng Principal Component Analysis

Sau khi dữ liệu được chuẩn hóa, Principal Component Analysis (PCA) được áp dụng nhằm giảm số chiều không gian đặc trưng trong khi vẫn giữ lại phần lớn thông tin quan trọng của dữ liệu.

Chuẩn hóa dữ liệu đầu vào

Cho tập dữ liệu huấn luyện:

$$X = [x_1, x_2, \ldots, x_N]^T \in \mathbb{R}^{N \times d},$$

trong đó mỗi $x_i \in \mathbb{R}^{67\,500}$.

Trước khi thực hiện PCA, dữ liệu được chuẩn hóa theo từng chiều đặc trưng:

$$\tilde{X} = \frac{X - \mu}{\sigma},$$

với $\mu \in \mathbb{R}^d$ và $\sigma \in \mathbb{R}^d$ lần lượt là vector trung bình và độ lệch chuẩn được tính trên tập huấn luyện. Bước này đảm bảo các đặc trưng có cùng thang đo và giúp PCA hoạt động ổn định hơn về mặt số học.

Phân rã giá trị suy biến (SVD)

Thay vì trực tiếp tính ma trận hiệp phương sai, PCA trong nghiên cứu này được triển khai thông qua phân rã giá trị suy biến (Singular Value Decomposition – SVD).

Cụ thể, thực hiện phân rã:

$$\tilde{X} = U \Sigma V^T,$$

trong đó:

- $U \in \mathbb{R}^{N \times N}$ là ma trận trực chuẩn trái,
- $\Sigma \in \mathbb{R}^{N \times d}$ là ma trận đường chéo chứa các giá trị suy biến σ_i,
- $V \in \mathbb{R}^{d \times d}$ là ma trận trực chuẩn phải.

Các vector hàng của V^T tương ứng với các vector riêng của ma trận hiệp phương sai

$$\Sigma_X = \frac{1}{N-1} \tilde{X}^T \tilde{X}.$$

Giá trị riêng và phương sai giữ lại

Các trị riêng của Σ_X được xác định từ các giá trị suy biến:

$$\lambda_i = \frac{\sigma_i^2}{N-1}.$$

Tỷ lệ phương sai được giải thích bởi thành phần chính thứ i được xác định bởi:

$$\mathrm{EVR}_i = \frac{\lambda_i}{\sum_{j=1}^{d} \lambda_j}.$$

Trong thực nghiệm, $k = 200$ thành phần chính đầu tiên được lựa chọn sao cho tổng phương sai giữ lại:

$$\sum_{i=1}^{k} \mathrm{EVR}_i$$

đạt giá trị đủ lớn, đảm bảo dữ liệu sau giảm chiều vẫn giữ được cấu trúc quan trọng của dữ liệu ban đầu.

Phép chiếu dữ liệu

Gọi:

$$W_k = [v_1, v_2, \ldots, v_k]^T \in \mathbb{R}^{k \times d}$$

là ma trận gồm k vector riêng tương ứng với k trị riêng lớn nhất.

Dữ liệu sau khi giảm chiều được xác định bởi:

$$Z = \tilde{X} W_k^T \in \mathbb{R}^{N \times k}.$$

Như vậy, không gian đặc trưng được thu gọn từ:

$$\mathbb{R}^{67\,500} \longrightarrow \mathbb{R}^{200}.$$

Phép biến đổi PCA sau khi được huấn luyện trên tập huấn luyện được áp dụng đồng nhất cho cả tập huấn luyện và tập kiểm thử, đảm bảo không xảy ra hiện tượng rò rỉ thông tin (*data leakage*).

Bước giảm chiều này giúp giảm đáng kể chi phí tính toán, tăng độ ổn định của quá trình huấn luyện, và tạo điều kiện thuận lợi cho việc áp dụng vào mô hình Non-Linear SVM.

3.4.3 Huấn luyện mô hình Non-Linear Support Vector Machine

Sau khi dữ liệu đã được giảm chiều bằng PCA, mô hình Non-Linear Support Vector Machine được sử dụng để học ranh giới quyết định giữa hai lớp Shells và Pebbles.

Ánh xạ đặc trưng phi tuyến

Trong không gian đặc trưng sau PCA, dữ liệu vẫn không đảm bảo khả năng phân tách tuyến tính. Do đó, thay vì sử dụng Linear SVM, một ánh xạ phi tuyến được áp dụng để mở rộng không gian đặc trưng. Cụ thể, với mỗi mẫu dữ liệu $x \in \mathbb{R}^k$, một ánh xạ đa thức bậc hai được sử dụng:

$$\phi(x) = [x, x^2].$$

Phép ánh xạ này cho phép mô hình nắm bắt các mối quan hệ phi tuyến bậc hai giữa các thành phần đặc trưng, tương đương với việc sử dụng kernel đa thức bậc hai nhưng được triển khai một cách tường minh. Sau khi mở rộng, dữ liệu được chuẩn hóa lại để đảm bảo các đặc trưng mới có cùng thang đo, giúp quá trình tối ưu ổn định hơn.

Hàm quyết định

Trong không gian đặc trưng mở rộng, hàm quyết định của mô hình có dạng:

$$f(x) = w^T \phi(x) - b,$$

trong đó w là vector trọng số và b là hệ số dịch chuyển.

Nhãn lớp được ánh xạ về:

$$y \in \{-1, +1\},$$

phù hợp với bài toán phân loại nhị phân của SVM.

Hàm mất mát và bài toán tối ưu

Mô hình được huấn luyện bằng cách tối ưu hàm mất mát dạng hinge loss kết hợp với điều chuẩn L_2:

$$L(w) = \lambda \|w\|^2 + \frac{1}{N} \sum_{i=1}^{N} \max\left(0, 1 - y_i f(x_i)\right),$$

trong đó:

- λ là tham số điều chuẩn,
- thành phần $\|w\|^2$ giúp kiểm soát độ phức tạp của mô hình,
- hinge loss khuyến khích việc mở rộng khoảng cách biên giữa hai lớp dữ liệu.

So với Linear SVM, sự khác biệt quan trọng nằm ở việc hàm $f(x)$ được xác định trên không gian đặc trưng phi tuyến, cho phép mô hình hóa các ranh giới quyết định có dạng cong và phức tạp hơn.

Thuật toán tối ưu

Bài toán tối ưu được giải bằng Stochastic Gradient Descent (SGD). Tại mỗi epoch, tốc độ học được điều chỉnh theo quy luật giảm dần:

$$\eta_t = \frac{\eta_0}{1 + \alpha t},$$

trong đó:

- η_0 là learning rate ban đầu,
- α là hệ số decay,
- t là số epoch hiện tại.

Với mỗi mẫu huấn luyện (x_i, y_i), cập nhật tham số được thực hiện theo hai trường hợp:

Trường hợp 1: Mẫu được phân loại đúng và nằm ngoài biên

$$y_i f(x_i) \geq 1,$$

khi đó chỉ cập nhật điều chuẩn:

$$w \leftarrow w - \eta_t(2\lambda w).$$

Trường hợp 2: Mẫu vi phạm điều kiện biên

$$y_i f(x_i) < 1,$$

khi đó gradient được cập nhật:

$$w \leftarrow w - \eta_t(2\lambda w - y_i \phi(x_i)),$$

$$b \leftarrow b - \eta_t y_i.$$

Cách cập nhật này tương ứng trực tiếp với đạo hàm của hàm hinge loss trong bài toán soft-margin SVM.

Vì sao Non-Linear SVM vượt trội hơn Linear SVM

Về mặt toán học, Linear SVM chỉ có thể học được ranh giới quyết định dạng siêu phẳng trong không gian đặc trưng ban đầu:

$$f_{\text{linear}}(x) = w^T x - b.$$

Trong khi đó, Non-Linear SVM học siêu phẳng trong không gian đặc trưng mở rộng:

$$f_{\text{nonlinear}}(x) = w^T \phi(x) - b,$$

tương đương với việc học ranh giới phi tuyến trong không gian ban đầu.

Đối với dữ liệu ảnh, đặc trưng pixel và các thành phần PCA thường có quan hệ phi tuyến phức tạp. Do đó, Linear SVM có xu hướng*underfitting*, trong khi Non-Linear SVM có khả năng biểu diễn tốt hơn các cấu trúc phân biệt giữa Shells và Pebbles.

Thực nghiệm cho thấy, việc sử dụng ánh xạ đa thức bậc hai giúp cải thiện đáng kể khả năng phân tách lớp, dẫn đến độ chính xác cao hơn so với mô hình tuyến tính khoảng 3-5%.

Chương 4

Thực nghiệm và đánh giá mô hình

4.1 Thiết lập thí nghiệm

Thực nghiệm được tiến hành nhằm đánh giá hiệu quả của mô hình Non-Linear Support Vector Machine khi kết hợp với phương pháp giảm chiều PCA trên tập dữ liệu Shells & Pebbles.

Dữ liệu đầu vào đã được tiền xử lý và giảm chiều theo pipeline được trình bày trong Chương 3. Toàn bộ thí nghiệm được thực hiện trên tập dữ liệu đã được chia sẵn thành:

- Tập huấn luyện: 80%
- Tập kiểm thử: 20%

Việc chia tập được cố định seed ngẫu nhiên nhằm đảm bảo tính tái lập của kết quả thực nghiệm.

4.2 Thiết lập tham số huấn luyện

Sau bước PCA, dữ liệu được giảm từ không gian ban đầu xuống còn:

$$k = 200$$

chiều đặc trưng chính.

Mô hình Non-Linear SVM được huấn luyện với các tham số sau:

- Learning rate ban đầu: $\eta_0 = 2 \times 10^{-6}$
- Hệ số điều chuẩn: $\lambda = 5 \times 10^{-4}$
- Số epoch huấn luyện: 3000
- Hệ số giảm learning rate: $\alpha = 0.001$
- Ánh xạ đặc trưng: đa thức bậc hai $\phi(x) = [x, x^2]$

Nhãn lớp được ánh xạ về dạng:

$$y \in \{-1, +1\},$$

trong đó:

- -1: Pebbles
- $+1$: Shells

Quá trình huấn luyện được thực hiện bằng Stochastic Gradient Descent, đồng thời theo dõi giá trị hinge loss để đánh giá mức độ hội tụ của mô hình.

4.3 Chỉ số đánh giá

Hiệu năng mô hình được đánh giá thông qua hai chỉ số chính:

4.3.1 Độ chính xác (Accuracy)

Độ chính xác được định nghĩa là tỷ lệ số mẫu được phân loại đúng trên tổng số mẫu:

$$\text{Accuracy} = \frac{\text{Số dự đoán đúng}}{\text{Tổng số mẫu}}.$$

Chỉ số này phản ánh mức độ tổng quát của mô hình trên tập kiểm thử.

4.3.2 Ma trận nhầm lẫn

Ma trận nhầm lẫn được sử dụng để phân tích chi tiết khả năng phân loại của mô hình trên từng lớp dữ liệu. Với quy ước:

- True Positive (TP): Shells được phân loại đúng
- True Negative (TN): Pebbles được phân loại đúng
- False Positive (FP): Pebbles bị phân loại nhầm thành Shells
- False Negative (FN): Shells bị phân loại nhầm thành Pebbles

Ma trận nhầm lẫn có dạng:

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}.$$

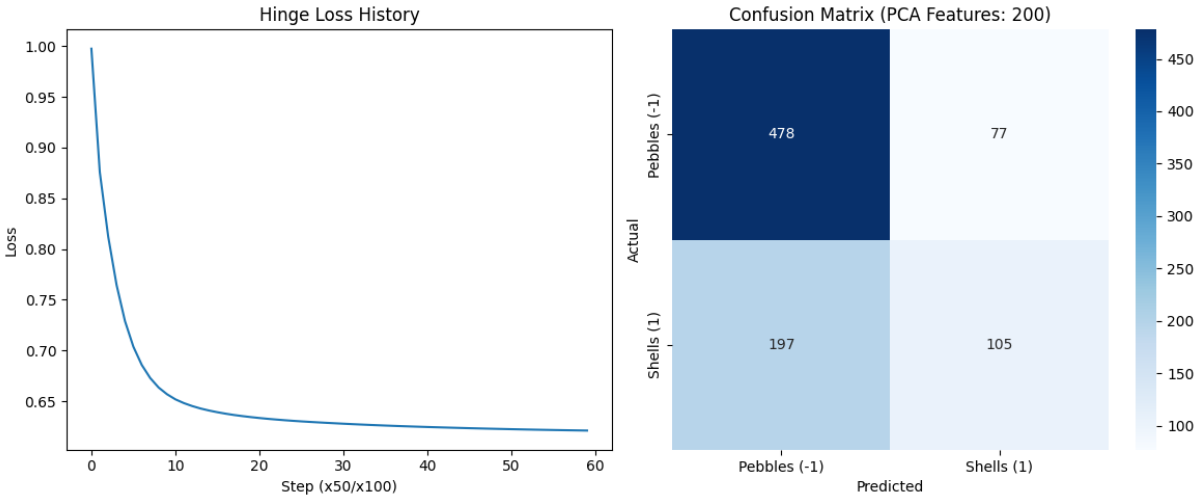
4.4 Kết quả thực nghiệm

Sau khi huấn luyện và đánh giá trên tập kiểm thử, sử dụng Google Colab với GPU T4 (16GB RAM), mô hình đạt được kết quả định lượng:

$$\text{Accuracy} = 68.03\%.$$

Đây có thể coi là một kết quả tương đối tốt, nếu so sánh với tập dữ liệu khổng lồ, shells và pebbles vốn là những vật thể khó phân biệt, cộng với việc chịu giới hạn về tài nguyên tính toán.

Hình 4.1 minh họa quá trình hội tụ của hinge loss trong quá trình huấn luyện và ma trận nhầm lẫn thu được trên tập kiểm thử.



Hình 4.1: Kết quả thực nghiệm

Đồ thị hinge loss cho thấy giá trị mất mát giảm nhanh ở các epoch đầu và dần ổn định về sau, phản ánh quá trình học ổn định và khả năng hội tụ của mô hình.

Ma trận nhầm lẫn cho thấy:

- Mô hình phân loại tốt lớp Pebbles với số lượng dự đoán đúng cao (TN lớn).
- Lớp Shells có tỷ lệ nhầm lẫn cao hơn, thể hiện qua số lượng FN đáng kể.

4.5 Nhận xét và phân tích kết quả

Kết quả thực nghiệm cho thấy việc kết hợp PCA và Non-Linear SVM giúp mô hình học được ranh giới quyết định phi tuyến trong không gian đặc trưng đã giảm chiều.

Tuy nhiên, độ chính xác chưa đạt mức cao do một số nguyên nhân chính:

- Đặc trưng pixel thông qua PCA chưa khai thác tốt thông tin hình dạng và texture phức tạp của Shells
- Dữ liệu hai lớp có sự chồng lấn lớn về mặt thị giác
- PCA tối ưu phương sai toàn cục, không tối ưu trực tiếp cho khả năng phân biệt lớp

Mặc dù vậy, kết quả đạt được vẫn cho thấy Non-Linear SVM là một phương pháp khả thi khi không sử dụng các mô hình học sâu, đồng thời minh họa rõ ràng vai trò của ánh xạ phi tuyến trong bài toán phân loại ảnh.

4.6 Phân tích ưu – nhược điểm và so sánh phương pháp

4.6.1 Ưu điểm của phương pháp Non-Linear SVM kết hợp PCA

Phương pháp Non-Linear SVM kết hợp với PCA có một số ưu điểm nổi bật trong bối cảnh bài toán phân loại ảnh với kích thước dữ liệu vừa phải:

- **Cơ sở lý thuyết vững chắc:** SVM được xây dựng trên nguyên lý tối ưu hóa biên (maximum margin), giúp mô hình có khả năng tổng quát hóa tốt ngay cả khi số chiều đặc trưng lớn.
- **Khả năng học ranh giới phi tuyến:** Thông qua ánh xạ đặc trưng đa thức, Non-Linear SVM có thể mô hình hóa các ranh giới quyết định phi tuyến mà các mô hình tuyến tính không thể biểu diễn.
- **Giảm độ phức tạp tính toán nhờ PCA:** PCA giúp giảm mạnh số chiều đặc trưng (từ hàng chục nghìn chiều pixel xuống vài trăm chiều), từ đó làm giảm thời gian huấn luyện và hạn chế hiện tượng overfitting.
- **Phù hợp với tập dữ liệu quy mô vừa:** Với số lượng ảnh ở mức vài nghìn, phương pháp này đạt được sự cân bằng hợp lý giữa hiệu năng và chi phí tính toán, không yêu cầu tài nguyên phần cứng lớn.

4.6.2 So sánh với các mô hình hồi quy (Regression-based Models)

Các mô hình hồi quy tuyến tính hoặc logistic regression thường được xem như baseline cho bài toán phân loại.

- Regression giả định ranh giới quyết định tuyến tính, do đó khó có thể phân tách hiệu quả dữ liệu ảnh có cấu trúc phức tạp.
- So với regression, Non-Linear SVM có khả năng biểu diễn ranh giới phân lớp linh hoạt hơn, đặc biệt khi dữ liệu không phân tách tuyến tính trong không gian đặc trưng.
- Trong bối cảnh bài toán này, regression có thể được xem là baseline đơn giản, nhưng thường cho độ chính xác thấp hơn đáng kể so với Non-Linear SVM sau khi áp dụng PCA.

4.6.3 So sánh với các mô hình học sâu (Deep Learning)

Mạng nơ-ron học sâu, đặc biệt là Convolutional Neural Networks (CNN), là lựa chọn phổ biến cho bài toán phân loại ảnh.

- CNN có khả năng tự động học các đặc trưng hình học và texture phức tạp, điều mà biểu diễn pixel thô kết hợp PCA không làm được.
- Tuy nhiên, CNN thường yêu cầu:
  - Tập dữ liệu lớn hơn để tránh overfitting
  - Tài nguyên tính toán cao (GPU)
  - Quy trình huấn luyện và tinh chỉnh phức tạp
- Trong bối cảnh dữ liệu hạn chế và mục tiêu nghiên cứu mang tính học thuật, Non-Linear SVM + PCA là lựa chọn hợp lý hơn nhờ tính đơn giản, dễ kiểm soát và dễ phân tích toán học.

4.6.4 So sánh với các phương pháp học kết hợp (Ensemble Learning)

Các phương pháp học kết hợp như Random Forest, Gradient Boosting hoặc kết hợp nhiều mô hình phân loại có thể cải thiện độ chính xác trong một số trường hợp.

- Ensemble có khả năng giảm phương sai và tăng độ ổn định mô hình, nhưng thường làm giảm khả năng diễn giải.
- Với đặc trưng ảnh có số chiều lớn, ensemble dựa trên cây quyết định có thể gặp khó khăn về bộ nhớ và thời gian huấn luyện.
- Ngược lại, SVM kết hợp PCA giữ được mô hình gọn nhẹ, có ý nghĩa hình học rõ ràng và dễ phân tích hành vi học của mô hình.

4.6.5 Nhận xét tổng quát

Tổng hợp lại, Non-Linear SVM kết hợp PCA không phải là phương pháp tối ưu tuyệt đối cho mọi bài toán phân loại ảnh, nhưng là một giải pháp hiệu quả và hợp lý trong bối cảnh dữ liệu vừa phải, tài nguyên tính toán hạn chế và yêu cầu cao về tính minh bạch và khả năng giải thích mô hình.

Phương pháp này cũng đóng vai trò như một baseline mạnh, làm nền tảng so sánh cho các mô hình học sâu hoặc các phương pháp trích xuất đặc trưng phức tạp hơn trong các nghiên cứu tiếp theo.

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Báo cáo đã xây dựng và đánh giá một pipeline phân loại ảnh dựa trên sự kết hợp giữa Principal Component Analysis (PCA) và Non-Linear Support Vector Machine (SVM) trên tập dữ liệu *Shells or Pebbles*. Toàn bộ quá trình được triển khai thủ công nhằm làm rõ cơ sở toán học của từng bước trong mô hình.

Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác

$$\text{Accuracy} = 68.03\%$$

trên tập kiểm thử. Việc sử dụng PCA giúp giảm đáng kể số chiều dữ liệu, qua đó làm cho việc huấn luyện Non-Linear SVM khả thi và ổn định hơn về mặt tính toán.

Phân tích ma trận nhầm lẫn cho thấy mô hình phân loại tốt hơn đối với lớp Pebbles, trong khi lớp Shells có tỷ lệ nhầm lẫn cao hơn. Nguyên nhân chủ yếu đến từ việc biểu diễn ảnh dựa trên pixel sau PCA chưa mô tả đầy đủ các đặc điểm hình dạng và cấu trúc bề mặt phức tạp của Shells.

5.2 Hướng phát triển

Từ những kết quả đạt được, một số hướng phát triển có thể được xem xét:

- Sử dụng các đặc trưng giàu thông tin hơn (HOG, LBP, hoặc đặc trưng trích xuất từ CNN) thay cho pixel thô.
- Kết hợp PCA với các phương pháp giảm chiều có giám sát hoặc các biến thể kernel để tăng khả năng phân biệt lớp.
- So sánh và kết hợp với các mô hình học sâu, đặc biệt là các mạng CNN nhẹ hoặc mô hình pretrained, nhằm đánh giá giới hạn của phương pháp hiện tại.

Nhìn chung, PCA kết hợp Non-Linear SVM là một baseline hợp lý cho bài toán phân loại ảnh khi tài nguyên tính toán hạn chế, đồng thời tạo nền tảng cho các hướng tiếp cận nâng cao hơn.

Tài liệu tham khảo

[1] T. Q. Long and T. V. Cường, *Giáo trình Học máy thống kê*. Hà Nội, Việt Nam: Nhà xuất bản Đại học Quốc gia Hà Nội.

[2] Chip Huyen, *Designing Machine Learning Systems*, O'Reilly Media, 2020.

[3] GeeksforGeeks, "Non-Linear Support Vector Machine," [Online]. Available: https://www.geeksforgeeks.org/machine-learning/ml-non-linear-svm/

[4] GeeksforGeeks, "Principal Component Analysis (PCA)," [Online]. Available: https://www.geeksforgeeks.org/data-analysis/principal-component-analysis-pca/

[5] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 1005–1009, 2013.

[6] freeCodeCamp, "SVM Kernels: How to Tackle Nonlinear Data in Machine Learning," [Online]. Available: https://www.freecodecamp.org/news/svm-kernels-how-to-tackle-nonlinear-data-in-machine-learning/