

Để cải tiến hiệu suất phân lớp văn bản tiếng Việt bằng cách kết hợp **Topological Data Analysis (TDA)** với kỹ thuật **tăng cường dữ liệu (data augmentation)** và **học sâu (deep learning)**, tôi sẽ trình bày một ví dụ chi tiết, dễ hiểu, dựa trên các đặc trưng topological (như trong file `features_calculation_by_thresholds.ipynb`) và áp dụng vào bài toán phân lớp văn bản (ví dụ: phân loại cảm xúc tích cực/tiêu cực). Tôi sẽ giải thích từng bước, bao gồm mục đích của TDA, cách tăng cường dữ liệu, và cách tích hợp vào mô hình học sâu, cùng với code minh họa.

Bài toán ví dụ

- **Mục tiêu:** Phân loại cảm xúc của các bình luận tiếng Việt về sản phẩm (tích cực vs. tiêu cực).
- **Dữ liệu:** Bộ dữ liệu gồm 5,000 bình luận tiếng Việt (tương tự `test_5k` trong file của bạn), ví dụ:
 - Tích cực: "Sản phẩm rất tốt, pin bền, camera đẹp!"
 - Tiêu cực: "Máy nhanh nóng, hiệu năng kém, thất vọng."
- **Thách thức:** Dữ liệu tiếng Việt thường ít, dẫn đến mô hình học sâu dễ bị overfitting. Cần tăng cường dữ liệu và tận dụng cấu trúc topological để cải thiện hiệu suất.

Cách tiếp cận

1. Tăng cường dữ liệu (Data Augmentation):

- Sử dụng các kỹ thuật như thay thế từ đồng nghĩa, dịch ngược (back-translation), hoặc thêm nhiễu để tạo thêm mẫu văn bản tiếng Việt.
- Mục đích: Tăng số lượng và sự đa dạng của dữ liệu, giúp mô hình khái quát hóa tốt hơn.

2 Trích xuất đặc trưng Topological:

- Sử dụng attention weights từ mô hình BERT tiếng Việt (như PhoBERT) để xây dựng đồ thị và tính các đặc trưng topological (H_0 , H_1 , Betti numbers, số cạnh, v.v.).
- Mục đích: Khai thác cấu trúc ngữ nghĩa ẩn (nhóm từ, chu trình) để bổ sung thông tin cho mô hình học sâu.

3 Kết hợp với học sâu:

- Tích hợp đặc trưng topological (như Betti numbers, số cạnh) vào mô hình học sâu (BERT + MLP hoặc CNN) để cải thiện hiệu suất phân lớp.
 - Mục đích: Kết hợp đặc trưng topological với đặc trưng ngữ nghĩa từ BERT để tăng độ chính xác và giảm overfitting.
-

Các bước chi tiết

1. Tăng cường dữ liệu (Data Augmentation)

• Kỹ thuật:

- **Thay thế từ đồng nghĩa:** Sử dụng từ điển đồng nghĩa tiếng Việt (hoặc công cụ như VnCoreNLP) để thay thế từ, ví dụ: "tốt" → "tuyệt vời".
- **Back-translation:** Dịch bình luận sang tiếng Anh và dịch ngược lại tiếng Việt, tạo biến thể mới.
- **Thêm nhiễu:** Thêm/tách từ ngẫu nhiên để tạo sự đa dạng.

• **Mục đích:** Tăng số lượng dữ liệu từ 5,000 lên 10,000 mẫu, giúp mô hình học sâu học được các đặc trưng ngữ nghĩa đa dạng hơn.

• Code minh họa:



data_augmentation.py
python

Show inline

- **Kết quả:** Từ 5,000 bình luận, tạo ra 15,000 bình luận (mỗi mẫu gốc sinh thêm 2 mẫu mới).

2. Trích xuất đặc trưng Topological

- **Phương pháp:**
 - Sử dụng **PhoBERT** (mô hình BERT cho tiếng Việt) thay vì bert-base-uncased để trích xuất attention weights.
 - Xây dựng đồ thị từ attention weights (như trong file gốc) và tính các đặc trưng topological (H_0, H_1 , Betti numbers, số cạnh) bằng Ripser hoặc GUDHI.
 - Các đặc trưng này phản ánh cấu trúc ngữ nghĩa (nhóm token, chu trình) trong văn bản.
- **Mục đích:** Bổ sung đặc trưng topological vào vector đặc trưng từ PhoBERT, giúp mô hình hiểu sâu hơn về cấu trúc văn bản tiếng Việt.
- **Code minh họa** (mở rộng từ file gốc):



topo_features.py
python

Show inline

- Kết quả:

- Tạo mảng `topo_features` chứa Betti-0, Betti-1, và số cạnh (`e`) cho mỗi mẫu, layer, head, và ngưỡng.
- Ví dụ: `topo_features[0, 0, 0, :, :]` chứa `[b0, b1, e]` cho layer 0, head 0, ngưỡng 0.025.

3. Kết hợp với học sâu

- Phương pháp:

- Sử dụng PhoBERT để trích xuất vector đặc trưng ngữ nghĩa (CLS token).
- Kết hợp với đặc trưng topological (Betti numbers, số cạnh) bằng cách nối (concatenate) hoặc đưa vào một MLP/CNN.
- Huấn luyện mô hình phân lớp (BERT + MLP) trên dữ liệu tăng cường.
- **Mục đích:** Đặc trưng topological cung cấp thông tin cấu trúc bổ sung, giúp mô hình phân biệt tốt hơn giữa văn bản tích cực và tiêu cực.

- Code minh họa:



`deep_learning_with_topo.py`
python

Show inline

- Kết quả:

- Mô hình kết hợp đặc trưng topological (Betti-0, Betti-1, số cạnh) với đặc trưng ngữ nghĩa từ PhoBERT.
- Độ chính xác tăng (ví dụ: từ 85% lên 90%) nhờ dữ liệu tăng cường và thông tin cấu trúc từ TDA.

Tại sao cải tiến hiệu suất?

1 Tăng cường dữ liệu:

- Tăng số lượng mẫu giúp mô hình học sâu khai quát hóa tốt hơn, đặc biệt với tiếng Việt (ngôn ngữ ít tài nguyên).
- Các biến thể văn bản (từ đồng nghĩa, back-translation) làm phong phú ngữ nghĩa, giảm overfitting.

2 Đặc trưng topological:

- **Betti-0:** Xác định các nhóm token liên quan (ví dụ: các từ tích cực như "tốt", "tuyệt vời" tạo thành một nhóm).
- **Betti-1:** Phát hiện các chu trình chú ý, giúp nhận diện mối quan hệ ngữ nghĩa phức tạp (như vòng lặp giữa các từ trong câu dài).
- **Số cạnh (e):** Phản ánh mức độ chú ý giữa các token, bổ sung thông tin về mật độ kết nối.
- Các đặc trưng này cung cấp góc nhìn hình học, bổ sung cho đặc trưng ngữ nghĩa từ PhoBERT.

3 Kết hợp học sâu:

- Đặc trưng topological được nối với vector CLS của PhoBERT, tạo thành đầu vào phong phú hơn cho lớp phân lớp.
- MLP xử lý đặc trưng topological giúp mô hình học được các mẫu cấu trúc ẩn, cải thiện khả năng phân biệt văn bản tích cực/tiêu cực.

Kết quả mong đợi

- **Độ chính xác:** Tăng từ 85% (chỉ dùng PhoBERT) lên ~90% (kết hợp TDA và tăng cường dữ liệu).
 - **Độ bền:** Mô hình ít bị overfitting hơn nhờ dữ liệu tăng cường và thông tin cấu trúc từ TDA.
 - **Ví dụ đầu ra:**
 - Bình luận: "Sản phẩm rất tốt, pin bền, camera đẹp!" → Dự đoán: Tích cực (0.95).
 - Bình luận: "Máy nhanh nóng, hiệu năng kém." → Dự đoán: Tiêu cực (0.92).
-

Lưu ý

- **Thư viện:**
 - **Ripser/GUDHI:** Tính Betti numbers và số cạnh (trong `count_top_stats`).
 - **Transformers:** PhoBERT cho tiếng Việt.
 - **PyTorch:** Huấn luyện mô hình học sâu.
 - **VnCoreNLP:** Xử lý ngôn ngữ tiếng Việt (từ đồng nghĩa, phân đoạn từ).
- **Cải tiến thêm:**
 - Thêm đặc trưng như **Persistence Images** hoặc **Mapper Algorithm** để tăng thông tin topological.
 - Tinh chỉnh PhoBERT trên dữ liệu tiếng Việt trước khi trích xuất đặc trưng.

Nếu bạn muốn code chi tiết hơn cho một phần cụ thể (như `count_top_stats`) hoặc ví dụ thực tế với dữ liệu cụ thể, hãy cho tôi biết!