TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH KỸ THUẬT MÁY TÍNH

**PHÂN LỚP LÀN ĐIỆU CHÈO VÀ DÂN CA QUAN HỌ**

Sinh viên thực hiện: **Lại Văn Hải**

Lớp: IS2 – K58

Giáo viên hướng dẫn: PGS.TS **Trịnh Văn Loan**

HÀ NỘI 05-2018

**PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

1. Thông tin về sinh viên

Họ và tên sinh viên: Lại Văn Hải

Điện thoại liên lạc: 0166 455 2105 Email: hailv.hedspi@gmail.com

Lớp: IS2 Hệ đào tạo: Công nghệ thông tin Việt Nhật

Đồ án tốt nghiệp được thực hiện tại: trường đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 01/01/2018 đến 20/05/2018

2. Mục đích nội dung của ĐATN

Phân lớp làn điệu chèo và dân ca quan họ Bắc Ninh

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu khái quát về nhận dạng làn điệu âm nhạc

- Tìm hiểu mô hình GMM

- Tìm hiểu về bộ công cụ ALIZE, Spro.

- Sử dụng bộ công cụ ALIZE và Spro trong phân lớp làn điệu chèo và dân ca quan họ

4. Lời cam đoan của sinh viên:

Tôi Lại Văn Hải cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của PGS.TS.Trịnh Văn Loan

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 20 tháng 05 năm 2018*  Tác giả ĐATN  Lại Văn Hải |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Giáo viên hướng dẫn  PGS.TS. Trịnh Văn Loan |

LỜI CẢM ƠN

Để có thể hoàn thiện được Đồ án tốt nghiệp như ngày hôm nay, nếu chỉ nhờ sự cố gắng của riêng bản thân em thì không thể nào hoàn thành được. Lời đầu tiên em xin chân thành cảm ơn thầy Trịnh Văn Loan đã dìu dắt em qua ba kỳ đồ án, xuất phát điểm chỉ với nền tảng duy nhất là sự hứng thú với đề tài của thầy, ngoài ra thì không có bất kỳ kiến thức nào khác, thầy đã tạo điều kiện cho nhóm đồ án được tham gia những tiết học mà trong chương trình đào tạo của chúng em không có, những câu chuyện bên lề mỗi buổi học của thầy giúp em định hướng được rất nhiều điều, nó vô cùng đáng quý với lứa tuổi chập chững như chúng em đang bị rất nhiều thứ khác cám dỗ. Em xin cảm ơn thầy Chu Bá Thành đã nhiệt tình chỉ dạy những mảng kiến thức vô cùng quan trọng liên quan đến đồ án của em, nhờ có sự chỉ dạy của thầy mà rất nhiều vấn đề khúc mắc của em đã được giải đáp. Em xin gừi lời cảm ơn đến người thân và những bạn bè xung quanh em đã luôn ở bên ủng hộ và khích lệ tinh thần cho em vào những lúc em yếu lòng nhất.

Cuối cùng em xin cảm ơn ngôi trường Bách Khoa đã tôi rèn em trưởng thành như ngày hôm nay, nhiều người bảo Bách Khoa là khó là khổ nhưng với em chưa từng một lần em hối hận khi chọn ngôi trường này. Rồi những ngày hôm nay sẽ trở thành kỷ niệm, nhưng em tin rằng mình đã có những kỷ niệm đẹp nhất ở cuối quãng đời sinh viên này.

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Hiện nay sự bùng nổ về công nghệ thông tin đã giúp cho các dịch vụ trực tuyến phát triển mạnh mẽ, một trong số đó có thể kể tới dịch vụ nghe nhạc trực tuyến. Tuy nhiên, cùng với sự phát triển này là sự bùng nổ về dữ liệu dẫn tới việc quản lý, phân loại và truy xuất dữ liệu đòi hỏi phải có các công cụ hữu ích hỗ trợ cho công việc này chứ không chỉ đơn thuần làm thủ công. Xuất phát từ nhu cầu thực tiễn này thì việc tạo ra các công cụ hỗ trợ cho việc phân lớp là vô cùng cần thiết

Hiện nay trên thế giới đã có nhiều công trình nghiên cứu về phân loại các thể loại âm nhạc hiện đại như Pop, Rock, Ballad,…Trong đồ án tốt nghiệp này em sẽ trình bày về “**Phân lớp làn điệu chèo và dân ca quan họ**”.

Các vấn đề sẽ được đề cập trong đề tài:

MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc514432698)

[LỜI CẢM ƠN 3](#_Toc514432699)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 4](#_Toc514432700)

[MỤC LỤC 4](#_Toc514432701)

[DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT 7](#_Toc514432702)

[DANH MỤC HÌNH 8](#_Toc514432703)

[DANH MỤC BẢNG 9](#_Toc514432704)

[MỞ ĐẦU 10](#_Toc514432705)

[Lý do chọn đề tài 10](#_Toc514432706)

[Mục đích, phạm vi nghiên cứu 10](#_Toc514432707)

[Đối tượng nghiên cứu 11](#_Toc514432708)

[Phương pháp nghiên cứu 11](#_Toc514432709)

[Nhiệm vụ nghiên cứu 11](#_Toc514432710)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 12](#_Toc514432711)

[1.1. Tổng quan về phân lớp làn điệu chèo và dân ca quan họ 12](#_Toc514432712)

[1.1.1. Đặc trưng của làn điệu chèo và dân ca quan họ 12](#_Toc514432713)

[1.1.2. Phân lớp làn điệu chèo và dân ca quan họ là gì? 13](#_Toc514432714)

[1.1.3. Các ứng dụng của phân lớp làn điệu 13](#_Toc514432715)

[1.2. Một số phương pháp phân lớp thường dùng 13](#_Toc514432716)

[1.3. Phương pháp phân lớp làn điệu chèo và dân ca quan họ 13](#_Toc514432717)

[1.4. 14](#_Toc514432718)

[Hình 1.1. Mô hình phân cụm chèo và dân ca quan họ 14](#_Toc514432719)

[1.5. Tìm hiểu một số công trình nhận phân lớp âm nhạc đã có tại Việt Nam 15](#_Toc514432720)

[CHƯƠNG 2. CÁC CÔNG CỤ SỬ DỤNG TRONG ĐỂ TÀI 17](#_Toc514432721)

[2.1. Bộ công cụ ALIZE 17](#_Toc514432723)

[4.1. 17](#_Toc514432725)

[4.1.1. Nguồn gốc 17](#_Toc514432726)

[4.1.2. Giới thiệu về ALIZE 17](#_Toc514432727)

[Hình 2.1. Thành phần của gói công cụ ALIZE 18](#_Toc514432728)

[4.1.3. Bộ công cụ SPro 18](#_Toc514432729)

[4.1.4. Sử dụng SPro và ALIZE trong phân lớp làn điệu 19](#_Toc514432730)

[Hình 2.2. Sơ đồ sử dụng các công cụ của SPro và ALIZE trong phân lớp làn điệu 19](#_Toc514432731)

[Bảng 2.1: Mô tả tùy chọn sử dụng tiêu chuẩn hóa năng lượng 22](#_Toc514432732)

[Bảng 2.2: Mô tả tùy chọn sử dụng phát hiện tín hiệu tiếng nói 23](#_Toc514432733)

[Bảng 2.3: Mô tả tùy chọn chuẩn hóa đặc trưng 23](#_Toc514432734)

[Bảng 2.4: Mô tả tùy chọn chính sử dụng huấn luyện mô hình từ 24](#_Toc514432735)

[Bảng 2.5: Mô tả tùy chọn trong huấn luyện mô hình đích 25](#_Toc514432736)

[CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG THỬ NGHIỆM NHẬN DẠNG LÀN ĐIỆU CHÈO VÀ DÂN CA QUAN HỌ 27](#_Toc514432737)

[5.1. Sơ đồ tổng quan quá trình xây dựng hệ thống nhận dạng làn điệu chèo và dân ca quan họ 27](#_Toc514432739)

[Hình 3.1. Hệ thống phân lớp làn điệu chèo và dân ca quan họ 28](#_Toc514432747)

[5.2. Chuẩn bị dữ liệu 28](#_Toc514432748)

[5.3. Phân lớp làn điệu sử dụng ALIZE 31](#_Toc514432749)

[Hình 3.1. Các bước xây dựng hệ phân lớp sử dụng SPro & ALIZE 31](#_Toc514432750)

[5.3.1. Bước 1: Tạo thư mục làm việc 32](#_Toc514432751)

[5.3.2. Bước 2: Tính tham số MFCC 32](#_Toc514432752)

[5.3.3. Bước 3: Dò tìm năng lượng 33](#_Toc514432753)

[5.3.4. Bước 4: Phát hiện tiếng nói trong tín hiệu 33](#_Toc514432754)

[5.3.5. Bước 5: Chuẩn hóa các tham số của tín hiệu 33](#_Toc514432755)

[5.3.9. Bước 9: Chuẩn hóa TrainWorldInit 35](#_Toc514432756)

[5.3.10. Bước 10: Chuẩn hóa TrainWorldFinal 36](#_Toc514432757)

[5.3.12. Bước 12: Huấn luyện GMM 36](#_Toc514432758)

[5.3.14. Bước 14: Tiến hành thử nghiệm 38](#_Toc514432759)

[Chương 4. KẾT QUẢ THỬ NGHIỆM 39](#_Toc514432760)

[Chương 5. KẾT LUẬN 39](#_Toc514432761)

[5.1. Những vấn đề đã giải quyết trong Đồ án 39](#_Toc514432763)

[6.2. Hướng phát triển của đề tài 40](#_Toc514432764)

[TÀI LIỆU THAM KHẢO 41](#_Toc514432765)

[PHỤ LỤC 42](#_Toc514432766)

[A. Mô hình hỗn hợp Gauss 42](#_Toc514432767)

[Hình A.1. Hàm mật độ Gauss 43](#_Toc514432768)

[Hình A.2. Mô hình GMM 44](#_Toc514432769)

[Hình A.3. Hàm mật độ của GMM có 3 phân phối Gauss 44](#_Toc514432770)

[B. Ba bài toán cơ bản của HMM 47](#_Toc514432771)

[B.1. Bài toán thứ nhất: Đánh giá xác suất 47](#_Toc514432772)

[B.2. Thuật toán thứ hai: Tìm dãy trạng thái tối ưu 48](#_Toc514432773)

[B.3. Thuật toán thứ ba: Ước lượng tham số của mô hình 49](#_Toc514432774)

[C. Cấu trúc các gói trong thư viện LIA-RAL 51](#_Toc514432775)

[C.1. Cấu trúc file NormFeat.cfg 51](#_Toc514432776)

[C.2. Cấu trúc file NormFeat\_energy.cfg 52](#_Toc514432777)

[C.3. Cấu trúc file EnergyDetector.cfg 53](#_Toc514432778)

[C.4. Cấu trúc file TrainWorldInit.cfg 55](#_Toc514432779)

[C.5. Cấu trúc file TrainWorldFinal.cfg 56](#_Toc514432780)

[C.6. Cấu trúc gói TrainTarget 57](#_Toc514432781)

[C.7. Cấu trúc gói ComputeTest 57](#_Toc514432782)

[D. Code chương trình nhận dạng người nói tiếng Việt 58](#_Toc514432783)

DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| ***Chữ viết tắt*** | ***Chữ viết đầy đủ*** |
| **ANN** | Artificial Neural Network |
| **ASV** | Automatic Speaker Identification |
| **ASI** | Automatic Speaker Verification |
| **ASR** | Automatic Speaker Recognition |
| **DTW** | Dynamic Time Warping |
| **EM** | Expectance Maximization |
| **FSG** | Finite State Grammar |
| **GMM** | Gaussion Mixture Model |
| **HMM** | Hidden Markov Model |
| **MFCC** | Mel-Frequency Cepstral Coefficients |
| **LM** | Language Model |
| **UML** | Unified Modeling Language |
| **VQ** | Vector Quantization |

DANH MỤC HÌNH

[Hình 1.1. Hệ thống xác định người nói 15](#_Toc367523867)

[Hình 1.2. Mô hình xác định người nói 16](#_Toc367523868)

[Hình 1.3. Hệ thống xác minh người nói 17](#_Toc367523869)

[Hình 1.4. Mô hình xác minh người nói 17](#_Toc367523870)

[Hình 1.5. Các thành phần của một hệ thống nhận dạng người nói 19](#_Toc367523871)

[phụ thuộc từ khóa 19](#_Toc367523872)

[Hình 1.6. Hệ thống định danh người nói sử dụng giải thuật GMM 21](#_Toc367523873)

[Hình 1.7. Pha nhận dạng hệ nhận dạng người nói phụ thuộc từ khóa thay đổi 23](#_Toc367523874)

[Hình 1.8. Kết quả khảo sát độ chính xác nhận dạng người nói phụ thuộc độ dài từ khóa 24](#_Toc367523875)

[Hình 2.1. Thành phần của gói công cụ ALIZE 26](#_Toc367523876)

[Hình 2.2. Sơ đồ các công cụ sử dụng ALIZE trong nhận dạng người nói 28](#_Toc367523877)

[Hình 3.1. Hệ nhận dạng người nói phụ thuộc từ khóa tiếng Việt 42](#_Toc367523878)

[Hình 3.2. File wav biểu diễn tín hiệu phát âm từ khóa 0 đến 9 của người nói 45](#_Toc367523879)

[Hình 3.3. Các bước xây dựng hệ nhận dạng người nói sử dụng SPro & ALIZE 53](#_Toc367523880)

[Hình 3.4. Sơ đồ tổng quát xây dựng Forced alignment proces 57](#_Toc367523881)

[Hình 3.5. Biểu đồ thể hiện kết quả nhận dạng với Test 1 63](#_Toc367523882)

[Hình 3.6. Biểu đồ thể hiện kết quả nhận dạng với Test 2 64](#_Toc367523883)

[Hình 3.7. Biểu đồ thể hiện kết quả nhận dạng với các thử nghiệm 66](#_Toc367523884)

[Hình A.1. Hàm mật độ Gauss 72](#_Toc367523885)

[Hình A.2. Mô hình GMM 73](#_Toc367523886)

[Hình A.3. Hàm mật độ của GMM có 3 phân phối Gauss 74](#_Toc367523887)

DANH MỤC BẢNG

[Bảng 2.1: Mô tả tùy chọn sử dụng tiêu chuẩn hóa năng lượng 30](#_Toc367523888)

[Bảng 2.2: Mô tả tùy chọn sử dụng phát hiện tín hiệu tiếng nói 31](#_Toc367523889)

[Bảng 2.3: Mô tả tùy chọn chuẩn hóa đặc trưng 32](#_Toc367523890)

[Bảng 2.4: Mô tả tùy chọn chính sử dụng huấn luyện mô hình từ 33](#_Toc367523891)

[Bảng 2.5: Mô tả tùy chọn trong huấn luyện mô hình đích 34](#_Toc367523892)

[Bảng 2.6: Mô tả tùy chọn tính điểm số của mô hình GMM 36](#_Toc367523893)

[Bảng 3.1: Thông tin về dữ liệu 48 người nói 43](#_Toc367523894)

[Bảng 3.2: Kết quả nhận dạng người nói thử nghiệm với Test 1: sử dụng xác suất tiên nghiệm lấy từ mô hình HMM và GMM của người nói 63](#_Toc367523895)

[Bảng 3.3: Kết quả nhận dạng người nói thử nghiệm với Test 2: Xác suất tiên nghiệm lấy từ mô hình GMM và xác suất hậu nghiệm lấy từ mô hình HMM của từng người nói 64](#_Toc367523896)

[Bảng 3.4: Kết quả tổng hợp các thử nghiệm theo thời gian 65](#_Toc367523897)

[Bảng 3.5: Bảng tổng hợp kết quả nhận dạng các chương trình đã thử nghiệm 66](#_Toc367523898)

MỞ ĐẦU

Lý do chọn đề tài

Hiện nay đang trong giai đoạn của cuộc Cách mạng công nghệ 4.0, sự bùng nổ về thông tin đang diễn ra trên toàn cầu, chưa khi nào mà việc con người đặt câu hỏi và tìm kiếm câu trả lời lại dễ dàng như hiện nay. Nhưng đối với bất kỳ câu chuyện nào cũng tồn tại hai mặt của một vấn đề, sự bùng nổ về mặt dữ liệu dẫn tới nhu cầu về mặt quản lý cũng khắt khe hơn, một trong những nguồn dữ liệu mà em muốn nhắc tới đó chính là âm nhạc.

Âm nhạc là một trong những thông tin phổ biến nhất và trên các trang web về nghe nhạc trực tuyến có đến hàng triệu bản nhạc, điều này đặt ra một thách thức lớn trong việc tổ chức, truy xuất và tìm kiếm. Một trang web nghe nhạc mà chỉ liệt kê một cách đơn thuần danh sách các bài hát có trong hệ thống thì không thể coi đây là một trang web tốt được, chính vì vậy nó đòi hỏi phải được phân loại, sắp xếp theo tên ca sĩ, tên album, thể loại, năm phát hành,… Chính từ nhu cầu này đã nảy sinh ra yêu cầu về việc phân loại âm nhạc, nhưng điều gì sẽ xảy ra nếu chúng ra thực hiện điều này thủ công, có thể với số lượng vài trăm file âm thanh thì chưa có vấn đề gì xảy ra nhưng câu chuyện sẽ trở nên hoàn toàn khác khi con số này lên tới hàng triệu. Do đó phân loại nhạc tự động thành các loại khác nhau là một nhiệm vụ quan trọng để truy xuất và tổ chức các thư viện nhạc.

Là một người có tình yêu với văn hóa truyền thống nên em rất hứng thú với những giá trị văn hóa cổ truyền của dân tộc và nhạc cổ truyền cũng không phải là ngoại lệ. Chính vì vậy, cùng với nhu cầu thực tiễn được đề cập ở bên trên nên em đã quyết định lựa chọn đề tài “**Phân lớp làn điệu chèo và dân ca quan họ**”, làm để tài nghiên cứu của mình. Với mong muốn thông qua việc này có thể góp một nhỏ vào việc trùng hưng lại những làn điệu chèo và quan họ cổ, để không chỉ có thế hệ chúng em mà cả những thế hệ sau này vẫn sẽ tiếp tục duy trì ngọn lửa văn hóa truyền thống trong lòng mỗi người dân Việt Nam.

Mục đích, phạm vi nghiên cứu

* Tìm hiểu tổng quan về phân lớp làn điệu âm nhạc.
* Nghiên cứu mô hình GMM và bộ công cụ ALIZE, gói thư viện LIA-RAL để nhận diện làn điệu âm nhạc.
* Tìm hiểu công cụ Spro để trích chọn đặc trưng.

Đối tượng nghiên cứu

* Nghiên cứu tổng quan về nhận dạng thể loại âm nhạc.
* Nghiên cứu về mô hình hỗn hợp GMM.
* Nghiên cứu bộ công cụ ALIZE, công cụ Spro.

Phương pháp nghiên cứu

* Nghiên cứu lý thuyết về nhận dạng thể loại âm nhạc.
* Nghiên cứu sử dụng bộ công cụ ALIZE trên môi trường Linux trong phân lớp làn điệu âm nhạc.
* Nghiên cứu sử dụng công cụ Spro trên môi trường Linux trong trích chọn đặc trưng.
* Tập hợp các file âm thanh của chèo và quan họ, tiến hành thử nghiệm nhận dạng cho từng làn điệu một.

Nhiệm vụ nghiên cứu

* Tìm hiểu khái quát về nhận dạng làn điệu âm nhạc
* Tìm hiểu mô hình GMM.
* Tìm hiểu về bộ công cụ ALIZE, Spro.
* Sử dụng bộ công cụ ALIZE và Spro trong phân lớp làn điệu chèo và dân ca quan họ.

Nội dung đồ án được trình bày trong 105 trang và được chia thành 4 chương:

* Chương 1: Trình bày tổng quan về phân lớp làn điệu chèo và dân ca quan họ cùng với một số phương pháp phân lớp thường dùng. Chương này cũng trình bày các kết quả nghiên cứu liên quan đến phân loại thể loại âm nhạc tại Việt Nam.
* Chương 2: Trình bày về bộ công cụ ALIZE và SPro cùng với cách sử dụng những bộ công cụ này việc nghiên cứu thực tế.
* Chương 3: Chương này sẽ trình bày về cơ sở dữ liệu được sử dụng trong đề tài, cách triển khai hệ thống phân lớp với bộ công cụ ALIZE và SPro
* Chương 4: Trình bày kết quả thu được của đề tài.
* Chương 5: Nêu lên những vấn đề đã giải quyết trong đề tài và định hướng phát triển.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT PHÂN LỚP

* 1. Tổng quan về phân lớp
     1. Khái niệm

Ngày nay phân lớp dữ liệu là một trong những hướng nghiên cứu chính của khai phá dữ liệu, nhất là trong thời đại bùng nổ của big data đặt ra nhu cầu từ một cơ sở dữ liệu với nhiều thông tin ẩn con người có thể trích rút ra những thông tin có lợi để phục vụ cho các nghiệp vụ thông minh.

Phân lớp được hiểu là quá trình nhóm các đối tượng có những điểm tương đồng lại với nhau thành hữu hạn các nhóm. Có hai dạng bài toán phân lớp thường được đề cập đến trong học máy(ML – Machine Learning) đó là:

* **Phân lớp nhị phân(Binary Classification)**: được hiểu là việc phân các phần tử của một tập hợp nhất định thành hai nhóm dựa trên cơ sở của các quy tắc phân lớp. Một trong số các bài toán tiêu biểu của phân lớp nhị phân là xác định một bức thư điện tử có phải là thư rác hay không.
* **Phân lớp đa lớp(Multiclass Classification):** tương tự với logic của phân lớp nhị phân, tuy nhiên sự khác biệt lớn nhất của phân lớp đa lớp và phân lớp nhị phân đó chính là độ chính xác của bài toán bởi vì việc phân thành hai lớp sẽ dễ hơn rất nhiều so với việc phân thành nhiều lớp. Một số bài toán thực tế của phân lớp đa lớp là: nhận dạng khuôn mặt, nhận dạng chữ viết tay, nhận dạng ngôn ngữ,…

Việc giải quyết bài toán phân lớp đa lớp có thể được đơn giản hóa bằng việc chuyển thành nhiều bài toán phân lớp nhị phân. Ví dụ: bài toán nhận diện khuôn mặt thì chúng ta có thể chia thành hai bài toán nhỏ: bài toán một là nhận diện khuôn mặt là nam hay nữ, sau đó mới đi đến giải quyết bài toán thứ hai là nhận diện cụ thể khuôn mặt là của người nào. Thời gian thực hiện của việc chia nhỏ công việc này sẽ lâu hơn nhưng đem lại kết quả chính xác sẽ cao hơn do việc thu nhỏ phạm vi của bài toán.

* + 1. Các bước để xây dựng một hệ thống phân lớp

Để xây dựng một hệ thống phân lớp cần thực hiện qua hai bước sau đây:

* Bước 1: Xây dựng mô hình từ tập dữ liệu huấn luyện.
* Bước 2: Đánh giá mô hình, mục đích để kiểm tra tính đúng đắn của mô hình để từ đó có thể dùng nó để phân lớp dữ liệu mới.

Ở bước 1, đối với tập dữ liệu huấn luyện trước tiên sẽ được chuẩn hóa và lựa chọn những thông tin cần thiết liên quan đến bài toán phân lớp cần xử lý sau đó mỗi bộ dữ liệu sẽ được phân vào một lớp xác định trước(xác định bởi thuộc tính dán nhãn lớp), từ tập dữ liệu này sẽ xây dựng được mô hình dựa trên các luật phân lớp, các cây quyết định hoặc các công thức toán học.

Bước 2 sẽ tiến hành phân lớp cho những đối tượng mới hoặc chưa được phân lớp từ đó chúng ta có thể đánh giá được độ chính xác của mô hình dựa trên việc so sánh lớp biết trước của bộ dữ liệu dùng để kiểm tra với kết quả thu được từ mô hình.

* 1. Một số mô hình phân lớp thường dùng
     1. **Mô hình K-láng giềng gần nhất(K-Nearest Neighbors – K-NN)**

K-láng giềng gần nhất là một trong những thuật toán học có giám sát(Supervised learning) - là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Khi tiến hành huấn luyện thì thuật toán này không học một điều gì từ dữ liệu huấn luyện, mọi tính toán sẽ được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

Trước khi triển khai thuật toán cần phải chuẩn bị một tập dữ liệu huấn luyện – các dữ liệu này đều đã được dán nhãn phân lớp. Khi cần xác định một đối tượng chưa biết thuộc vào lớp nào, thuật toán sẽ được tiến hành thông qua các bước sau đây:

1. Xác định giá trị tham số K(số điểm láng giềng gần nhất)
2. Tính khoảng cách giữa đối tượng cần phân lớp với các điểm trong tập dữ liệu dùng để huấn luyện.
3. Lấy tất cả các lớp của K điểm gần nhất.
4. Trong các điểm láng giềng này lớp nào có các điểm chiếm phần lớn thì đối tượng cần phân lớp sẽ thuộc về lớp đó.

Cụ thể, kết quả phân lớp trả về chỉ dựa trên thông tin của K điểm dữ liệu trong tập huấn luyện gần nó nhất, không quan tâm đến việc có một vài điểm dữ liệu trong những điểm này là nhiễu(điều này dễ xảy ra với giá trị K nhỏ)

Ví dụ về việc triển khai thuật toán K-NN

Hình 1.1. Mô tả việc triển khai thuật toán K-NN

Tập dữ liệu huấn luyện của chúng ta có là hai lớp được ký hiệu là hình vuông và hình tam giác, đối tượng cần phân lớp là hình tròn. Bây giờ chúng ta sẽ đi tiến hành dự đoán lớp của đối tượng này dựa vào việc lựa chọn số láng giềng gần nhất với nó. Nhìn vào hình 1.1, ta xét với với một số giá trị K như sau:

* K = 1 – so sánh với 1 điểm láng giềng gần nhất thì đối tượng này sẽ thuộc lớp hình vuông do khoảng cách tới 1 điểm gần nhất là hình vuông.
* K = 2 – so sánh với 2 điểm láng giềng gần nhất thì sẽ không xác định được lớp mà đối tượng này thuộc về bởi vì 2 điểm gần nhất thì 1 điểm thuộc về lớp hình vuông và một điểm thuộc về lớp hình tam giác.
* K = 5 – so sánh với 5 điểm láng giềng gần nhất thì đối tượng này sẽ thuộc vào lớp tam giác, bởi trong 5 điểm gần nhất thì có 3 điểm thuộc lớp tam giác và 2 điểm thuộc lớp hình vuông nên chúng ta sẽ chọn lớp mà có số điểm chiếm ưu thế hơn.

Trong thực tế, khi sử dụng thuật toán K-NN để đạt được hiệu quả tốt nhất người ta thường thử nghiệm với các giá trị K khác nhau, sau đó tổng hợp kết quả và lựa chọn ra giá trị K đem lại kết quả thử nghiệm chính xác cao nhất. Bên cạnh việc thay đổi giá trị K thì cũng cần luân phiên thay kiểm thử với bộ dữ liệu huấn luyện và thử nghiệm khác nhau sau đó lấy kết quả trung bình, vì rất có thể dữ liệu phân chia trong một trường hợp cụ thể là rất tốt hoặc rất xấu.

Ưu điểm và nhược điểm của thuật toán K-NN

* Ưu điểm:
  + Độ phức tạp tính toán của quá trình huấn luyện bằng 0(do không có quá trình huấn luyện).
  + Việc dự đoán kết quả của dữ liệu mới đơn giản
* Nhược điểm:
  + Kết quả trả về dễ bị ảnh hưởng bởi nhiễu khi giá trị K nhỏ.
  + Vì mọi thao tác tính toán đều nằm ở giai đoạn thử nghiệm nên nếu với bộ dữ liệu huấn luyện lớn thì việc tính khoảng cách đến từng điểm dữ liệu sẽ gây ảnh hưởng đến thời gian chạy của chương trình.
    1. Mô hình cây quyết định(Decision Tree)

1. Khái niệm

Cây quyết định là một kiểu mô hình dự báo, đây là một dạng mô hình phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào tập hợp các luật.

Một cây quyết định bao gồm các thành phần sau:

* Các nút trong(internal node) hay nút không phải nút lá(non-leaf node) được dùng để biểu diễn một thuộc tính hay một câu hỏi. Các nút này thường có hai hay nhiều nút con, các nút con này có thể là một nút lá hoặc là một nút trong khác. Nếu tất cả các nút không phải nút lá đều có hai nút con thì ta nói rằng đó là một cây quyết định nhị phân(Binary Decision Tree).
* Nút lá(leaf node) biểu diễn các lớp quyết định.
* Nút không phải là nút lá mà ở trên cùng thì đó gọi là nút gốc(root node).
* Các nhánh(branch) biểu diễn các giá trị có thể có của thuộc tính hoặc câu trả lời cho câu hỏi được nêu ra ở nút cha.

Cây quyết định có thể được dùng để phân lớp bằng cách xuất phát từ gốc của cây và di chuyển theo các nhánh cho đến khi gặp nút lá. Trên cơ sở đó chúng ta có thể xây dựng các luật quyết định.

Cây quyết định là một mô hình học có giám sát, có thể được áp dụng vào cả hai bài toán phân lớp(classification) và hồi quy(regression – ước lượng kết quả trả về là số thực, ví dụ như ước tính giá của một ngôi nhà hay thời gian sửa chữa của một chiếc ô tô).

Ví dụ về bài toán phân lớp:

y

Đ

x < x1?

S

x > x2?

S

Đ

y1

y > y1?

S

Đ

0

x2

x1

x

Hình 1.2. Mô tả bài toán phân lớp sử dụng cây quyết định

Nhìn vào hình 1.2, ta có hai lớp hình vuông và hình tam giác trên mặt phẳng không gian hai chiều. Bài toán đặt ra ở đây là tìm ranh giới phân chia hai lớp này và xác định một điểm dữ liệu mới thuộc vào class nào, nhìn vào hình ta thây ranh giới của hai lớp này là giao nhau của 2 đường thằng x = x1, x = x2 và y = y3. Từ ranh giới này ta xây dựng nên các luật để xác định một điểm dữ liệu mới được đưa vào sẽ thuộc lớp nào. Ví dụ điểm mới này có tọa đồ (x0,y0), đối chiếu với sơ đồ cây quyết định bên phải ta có, nếu x0 < x1 thì ta xác định điểm này thuộc lớp hình tam giác, còn nếu không ta xét tiếp đến câu hỏi thứ 2, nếu x0 > x2 thì điểm này thuộc lớp hình tam giác, ngược lại thì ta lại xét đến câu hỏi thứ 3… Quá trình này được tiến hành liên tục cho đến khi xác định được lớp của đối tương thì dừng lại.

1. Xây dựng cây quyết định

Đứng trước một bài toán để tạo nên một cây quyết định chúng ta có hai giải pháp như sau:

* **Giải pháp 1:** Tham khảo ý kiến của các chuyên gia liên quan đến vấn đề cần xử lý. Đây là một giải pháp được đánh giá là hữu hiệu bởi độ chính xác cao của nó. Tuy nhiên, không phải lúc nào cũng có thể tìm được chuyên gia để giải quyết vấn đề mà chúng ta gặp phải.
* **Giải pháp 2:** Tạo ra các thuật toán để tự xây dựng cây quyết định, điều này được thực hiện dựa trên những bộ dữ liệu dùng để huấn luyện.

Việc xây dựng một cây quyết định bắt đầu từ gốc, tất cả các tập dữ liệu huấn luyện đều ở gốc, sau đó thuật toán sẽ tìm ra câu hỏi đầu tiên tốt nhất để bắt đầu sau đó phân chia các mẫu dựa trên các câu hỏi được lựa chọn. Sau mỗi câu hỏi, dữ liệu được phân chia vào từng nút con tương ứng với các câu trả lời cho câu hỏi đó. Câu hỏi ở đây chính là một thuộc tính, câu trả lời chính là giá trị của thuộc tính đó. Một phép phân chia tốt nhất là khi dữ liệu trong nút con hoàn toàn thuộc vào một lớp nào đó – lúc này thì nút con được coi như là một nút lá. Khi không còn đặt được thêm câu hỏi nào có giá trị nữa thì thuật toán sẽ dừng lại và kết thúc quá trình xây dựng cây quyết định. Hiệu quả phân lớp của cây quyết định phụ thuộc rất nhiều vào tập dữ liệu huấn luyện, vì vậy cần một tập dữ liệu đủ lớn và chính xác.

Sau khi xây dựng được cây quyết định thì trước khi đưa vào thử nghiệm cần chuẩn hóa cây bằng việc xác định và loại bỏ những nhánh nhiễu(những câu hỏi trả về kết quả không có nhiều giá trị).

Ưu điểm của cây quyết định:

* Đây là một mô hình dễ hình dung, gần với thực tế.
* Do tất cả các thông tin của dữ liệu đều có thể sử dụng để xây dựng nên các luật của cây quyết định nên việc chuẩn hóa dữ liệu là không cần thiết, điều này giúp giảm chi phí thiết kế hệ thống.
* Có thể đưa ra câu trả lời minh bạch về kết quả trả về của cây quyết định dựa vào tập hợp các luật.
* Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong khoảng thời gian ngắn.
  + 1. Mô hình Markov ẩn( Hidden Markov Model)

Học thuyết về chuỗi Markov được phát triển vào những năm 1990, được đặt theo tên của nhà toán học người Nga Andrei Andreyevich Markov. Mô hình Markov ẩn phát triển vào cuối những năm 60 và được sử dụng rộng rãi trong lĩnh vực nhận dạng tiếng nói vào những năm 1960 – 1970 và được đưa vào khoa học máy tính năm 1989.

1. Mô hình Markov

Xét một hệ thống bao gồm N trạng thái phân biệt, được đánh số thứ tự 1, 2, ..., N. Tại thời điểm t bất kỳ, hệ thống có thể chuyển từ trạng thái Si sang một trong N -1 trạng thái còn lại hoặc chuyển về chính trạng thái Si. Như vậy, ở tại thời điểm t, từ trạng thái Si có N nhánh thao tác chuyển trạng thái, mỗi một nhánh này có một xác suất xảy ra khác nhau và được gọi là **xác suất chuyển trạng thái.**

* Gọi S là tập hợp tất cả các trạng thái:

S = (s1, s2, …, sN) (1.1)

* Gọi qt là trạng thái đạt được ở thời điểm t, A là ma trận chuyển đổi chứa các giá trị aij là xác suất chuyển trạng thái từ si sang sj. Xác suất chuyển trạng thái aij không phụ thuộc vào thời gian t và độc lập với các trạng thái chuyển trước đó. Chỉ phụ thuộc duy nhất vào trạng thái hiện tại. Quá trình mang tính ngẫu nhiên này, được coi là “có thuộc tính Markov”

A = [aij], aij = P( qt = sj | qt – 1 = si) 1 ≤ i, j ≤ N (1.2)

Giữa mỗi bước thời gian, trạng thái tiếp theo được chọn một cách ngẫu nhiên. Trạng thái hiện tại sẽ quyết định xác suất phân bố của trạng thái tiếp theo. Nếu trạng thái tiếp theo chỉ phụ thuộc vào trạng thái hiện tại và không phụ thuộc trạng thái nào trong quá khứ thì đây được gọi là Mô hình Markov bậc 1.

Hình 1.3. Mô hình Markov bậc 1

P(qt | qt-1, qt-2, …) = P(qt | qt-1) (1.3)

Mô hình Markov bậc 2 là trạng thái tiếp theo phụ thuộc vào trạng thái hiện tại và trạng thái liền kề trước đó.

Hình 1.4. Mô hình Markov bậc 2

P(qt | qt-1, qt-2, …) = P(qt | qt-1, qt-2) (1.4)

1. Mô hình Markov ẩn

Mô hình Markov ẩn(HMM) là dạng mở rộng của mô hình Markov. Trong mô hình Markov, các sự kiện quan sát được nằm trong mỗi trạng thái và phụ thuộc vào hàm mật độ xác suất trong các trạng thái đó. Để dễ hình dung, chúng ta có thể xem hình 1.5

0.5

0.2

0.75

0.5

0.3

0.2

0.05

0.3

0.2

Hình 1.5. Mô hình Markov trong bài toán dự báo thời tiết

Mô hình Markov thực hiện tính toán trực tiếp trên các trạng thái còn mô hình Markov ẩn không tính toán kết quả trực tiếp trên các trạng thái đó mà phải thông qua một sự kiện trạng thái khác gọi là các sự kiện trạng thái quan sát.

v3

v1

v2

b23

b22

b21

a22

a12

v1

a11

b31

b13

v3

a32

a23

a21

b32

v2

b12

v2

a13

b33

v3

b11

v1

a31

a33

Hình 1.6. Mô hình Markov ẩn 3 trạng thái

Hình 1.6 mô tả mô hình Markov ẩn 3 trạng thái với các sự kiện có thể quan sát được trong mỗi trạng thái là V = (v1, v2, v3). Xác suất quan sát được sự kiện vk trong trạng thái sj phụ thuộc vào hàm xác suất bj(k). Hàm b được gọi là hàm mật độ xác suất của các sự kiện được quan sát.

Vẫn với bài toán dự báo thời tiết được đề cập ở hình 1.5, nếu xét trong mô hình Markov ẩn thì các trạng thái có thể quan sát được là: lầy lội, ẩm ướt, khô.

HMM được xác định bởi 5 thành phần:

* Tập hợp các trạng thái ẩn: N – số trạng thái

S = (s1, s2, s3, …, sN) (1.5)

* Tập hợp các quan sát: M – số các quan sát

V = (v1, v2, v3, …, vM) (1.6)

* Q là chuỗi các trạng thái có thể xảy ra, có chiều dài T

Q = (q1, q2, q3, …. qT) (1.7)

* Tương ứng với chuỗi các trạng thái có chuỗi các quan sát có thể quan sát được:

O = (o1, o2, o3, …, oT) (1.8)

* П là ma trận xác suất khởi tạo:

П = [πi], πi = P(q1 = si) 1 ≤ i ≤ N (1.9)

* A là ma trận chứa những giá trị xác suất chuyển đổi từ trạng thái i sang trạng thái j.

A = [aij], aij = P(qt = sj | qt-1 = si) 1 ≤ i, j ≤ N (1.10)

* B là ma trận xác suất quan sát, chứa những giá trị xác suất của quan sát k từ trạng thái i, độc lập với thời gian

B = [bi(k)], bi(k) = P(xt = vk | qt = si) 1 ≤ i ≤ N; 1 ≤ k ≤ M (1.11)

Tóm lại, các thành phần của HMM bao gồm:

* 2 tham số không đổi về kích cỡ: N và M(tổng số các trạng thái và các quan sát).
* 3 tập hợp phân bố xác suất: П, A, B. Một mô hình HMM được ký hiệu λ = (A, B, π)

1. Ứng dụng của HMM

* HMM được sử dụng nhiều trong phân tích ngôn ngữ: Nhận dạng tiếng nói( đối tượng quan sát được là âm thanh, đối tượng ẩn là từ ngữ).
* Nhận dạng chữ viết tay( đối tượng quan sát được là ký hiểu, đối tượng ẩn là từ ngữ).
* Phân loại từ ngữ (đối tượng quan sát được là từ ngữ, đối tượng ẩn là các loại từ: danh từ, động từ, tính từ)
* Hệ thống phiên dịch ngôn ngữ( đối tượng quan sát được là từ nước ngoài, đối tượng ẩn là từ ngữ ứng với ngôn ngữ cần dịch)
  + 1. Mô hình hỗn hợp Gauss

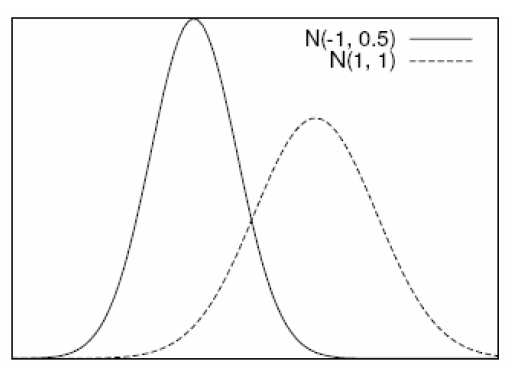
1. Đặc tả mô hình

Mô hình hợp Gauss (Gaussian Mixture Model - GMM) là một dạng mô hình thống

kê được xây dựng từ việc huấn luyện các tham số thông qua dữ liệu học. Mô hình

GMM còn có một số tên gọi khác như Weighted Normal Distribution Sums hay

Radial Basis Function Approximations…



Hình A.1. Hàm mật độ Gauss

Về cơ bản, mô hình GMM xấp xỉ một hàm mật độ xác suất bằng hợp các hàm mật

độ Gauss. Hình A.1 minh họa hai hàm mật độ Gauss với các tham số khác nhau.

Một cách hình thức, hàm mật độ xác suất của phân phối Gauss fN(x, μ, σ2) được cho bởi công thức:

 ( 1.1)

trong đó, μ là giá trị trung bình, σ là độ lệch chuẩn. Trong trường hợp x là vector

gồm D thành phần, hàm mật độ xác suất của phân phối Gauss fN(x, μ, Σ) được cho

bởi công thức:



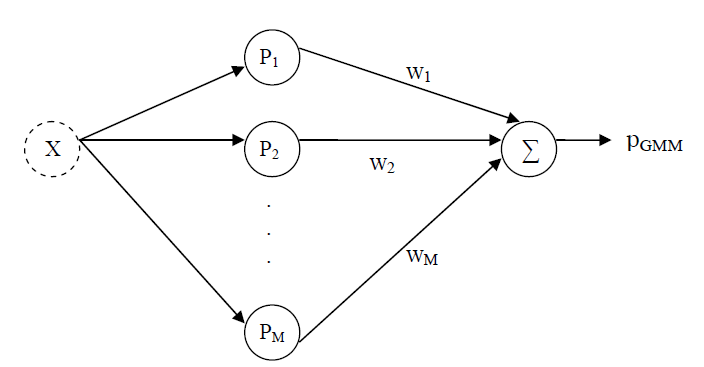
khi đó, μ là vector trung bình, Σ là ma trận hiệp phương sai. Nếu chọn μ=0 và σ=1,

công thức (1.1) sẽ trở thành hàm mật độ chuẩn Gauss:



Từ “Gauss” được đặt theo tên của nhà toán học người Đức Carl Friedrich Gauss.

Ông đã định nghĩa hàm mật độ Gauss và áp dụng trong phân tích dữ liệu thiên văn.



Hình A.2. Mô hình GMM

Cho trước M phân phối Gauss p1, p2, …, pM, hàm mật độ xác suất của mô hình

GMM được minh họa trong hình 3.2 chính là tổng trọng của M phân phối Gauss

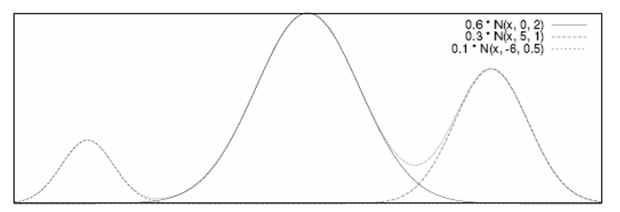
theo công thức:



trong đó, wi là trọng số của phân phối Gauss thứ i, thỏa ràng buộc 0≤ wi ≤1 và

 Các trọng số này thể hiện mức độ ảnh hưởng của mỗi phân phối Gaussđối với mô hình GMM. Như vậy, phân phối Gauss có phương sai và trọng số lớn bao nhiêu thì có mức độ ảnh hưởng lớn bấy nhiêu đối với kết xuất của mô hình.

Hình 3.3 cho thấy mức độ ảnh hưởng của từng phân phối Gauss lên GMM.



Hình A.3. Hàm mật độ của GMM có 3 phân phối Gauss

số λ = { wi, μi, Σi }, i ∈ [1, M]. Trong hướng tiếp cận GMM giải quyết bài toàn định danh người nói, mỗi người nói sẽ được mô hình hóa bằng một mô hình GMM mà bộ tham số λ của nó sẽ được xác định thông qua việc huấn luyện trên tập mẫu học của người nói tương ứng.

Tùy thuộc vào cách tổ chức của ma trận hiệp phương sai (Covariance Matrix), GMM có thể có một số biến thể khác nhau:

- Nodal covariance matrices GMM: mỗi phân phối Gauss trong GMM có một

ma trận hiệp phương sai riêng.

- Grand Covariance Matrix GMM: mọi phân phối Gauss trong một GMM dùng

chung một ma trận hiệp phương sai.

- Global Covariance Matrix GMM: mọi phân phối Gauss trong tất cả các GMM

dùng chung một ma trận hiệp phương sai.

Ngoài ra, xét về dạng thức, ma trận hiệp phương sai gồm hai loại: Full (dạng đầy đủ) và Diagonal (dạng ma trận đường chéo). Thông thường, dạng Nodal-Diagonal

Covariance Matrices GMM được sử dụng phổ biến nhất.

1. Bài toán ước lượng mật độ

Trong bộ phân loại dựa trên mô hình thống kê, việc ước lượng các tham số của mô hình được thực hiện thông qua huấn luyện trên một số lượng lớn các dữ liệu học. Mục tiêu của bước huấn luyện là nhằm tổng quát hóa, mô hình hóa những đặc điểm chung nhất của tập dữ liệu học. Đối với mô hình GMM, một trong những kỹ thuật xác định bộ tham số λ của nó được áp dụng khá phổ biến là thuật toán Expectation-Maximization (EM). Bản thân EM là một thuật toán tổng quát, đem lại các kết quả khác nhau đối với các mô hình khác nhau. Ngoài ra, có hai tiêu chí ước lượng khác nhau trong EM:

- Maximum Likelihood (ML): ước lượng tham số theo hướng cực đại hóa độ tương tự p(X | λ).

- Maximum A Posteriori Probability (MAP): ước lượng tham số theo hướng cực đại hóa xác suất quyết định p(λ | X).

Cho trước vector đặc trưng X trích được từ dữ liệu âm thanh, ta có thể dễ dàng tính được độ tương tự p(X | λ). Tuy nhiên, trong định danh người nói, vai trò quyết định lại nằm ở xác suất p(λ | X). Sử dụng công thức Bayes, ta có tương quan giữa p(X|λ) và p(λ|X):



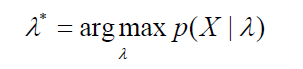
trong đó, p(X) là xác suất xuất hiện của vector đặc trưng X, p(λ) là tần suất xuất

hiện của người nói được mô hình hóa bởi GMM tương ứng. Trong luật quyết định

Bayes, p(X) độc lập; như vậy nếu giả định p(λ) là đồng nhất cho mọi người nói, ta

có thể quy vai trò quyết định từ p(λ | X) về p(X | λ) và áp dụng EM ước lượng λ

theo hướng maximum likelihood:



Như vậy, với tập dữ liệu huấn luyện X gồm T mẫu X  các trọng số, trung bình và phương sai của GMM ở mỗi bước lặp sẽ là:

Trọng số: 

Trung bình: 

Phương sai: 

trong đó,  là các thành phần tương ứng trong các vector  . Xác suất cho bởi công thức:



Trong quá trình xây dựng GMM, có hai vấn đề phát sinh là: số phân phối Gauss M

của mô hình, và bộ tham số khởi đầu λ0 trước khi tiến hành thuật toán EM. Hiện tại,

vẫn chưa có giải pháp tối ưu trên lý thuyết cho việc chọn M và λ0. Thông thường, M sẽ được chọn qua thực nghiệm, còn λ0 sẽ được khởi tạo bằng thuật toán K-means

nhằm đem lại khả năng cao hơn cho việc đạt tối ưu toàn cục, đồng thời đẩy nhanh

tốc độ hội tụ trong huấn luyện.

* 1. Tổng quan về phân lớp làn điệu chèo và dân ca quan họ
     1. Đặc trưng của làn điệu chèo và dân ca quan họ

1. Đặc trưng của làn điệu chèo

Hát chèo được hình thành và bắt nguồn từ các làn điệu dân ca, lời hát Chèo lấy trong các sáng tác văn học dân gian ở vùng đồng bằng Bắc Bộ là chủ yếu. Ngoài ra các làn điệu Chèo còn chịu những ảnh hưởng từ hát Văn, hát Xẩm, hát Ca trù, hát Xoan, hát Quan họ...

Chèo sử dụng tối thiểu là ba loại nhạc cụ dây là đàn nguyệt, đàn nhị và đàn bầu đồng thời thêm cả sáo nữa. Ngoài ra, các nhạc công còn sử dụng thêm trống và chũm chọe. Bộ gõ nếu đầy đủ thì có trống cái, trống con, trống cơm, thanh la, mõ. Trống con dùng để giữ nhịp cho hát, cho múa và đệm cho câu hát. Có câu nói "phi trống bất thành chèo" chỉ vị trí quan trọng của chiếc trống trong đêm diễn chèo. Trong chèo hiện đại có sử dụng thêm các nhạc cụ khác để làm phong phú thêm phần đệm như đàn thập lục, đàn tam thập lục, tiêu v.v...

1. Đặc trưng của làn điệu dân ca quan họ

Hát quan họ là hình thức hát đồng giọng: người hát dẫn, người hát luồn, hát đối đáp dẫn giọng, luồn giọng một cách điêu luyện. Giọng của hai người hát cặp với nhau phải tương đồng đến mức hai giọng trở thành một để tạo thành một âm thanh thống nhất.

Dân ca quan họ có 213 giọng khác nhau, với hơn 400 bài hát. Lời một bài hát có hai phần: lời chính và lời phụ. Lời chính là phần cốt lõi, phản ánh nội dung của bài hát, lời phụ gồm tất cả những tiếng nằm ngoài lời ca chính, là tiếng đệm, tiếng đưa hơi như i hi, ư hư, a ha, v.v…

Dân ca quan họ chủ yếu là nghệ thuật phổ lời ca dao và thơ. Nghệ thuật này đỏi hỏi phải sử dụng những tiếng phụ, lời phụ bên cạnh những tiếng chính, lời chính nhằm làm cho tiếng hát trôi chảy, bổ sung ý nghĩa cho lời hát chính, làm cho lời ca them phong phú, linh hoạt, tăng cường tính nhạc của bài ca, phát triển giai điệu, làm cho âm nhạc của bài ca trở nên sinh động, bố cục trở nên hợp lý. Không dùng tiếng phụ, lời phụ thì lời ca dễ đơn điệu, mất cân đối.

* + 1. Phân lớp làn điệu chèo và dân ca quan họ là gì?

Phân lớp được hiểu là quá trình nhóm các đối tượng lại với nhau dựa trên các điểm tương đồng. Bài toán đặt ra ở đây là phân lớp theo thể loại và phân lớp theo từng làn điệu cụ thể.

Phân lớp làn điệu là một hệ thống sinh trắc học, thực hiện việc tính toán xác định làn điệu dựa trên cơ sở sử dụng các đặc điểm đặc trưng của mỗi làn điệu. Đặc trưng của mỗi làn điệu chứa các thông tin sau:

* Thông tin mức thấp: độ cao, cường độ, tần số, quang phổ,… những đặc điểm này thường được nhận dạng bởi hệ thống phân lớp qua quá trình huấn luyện để đưa ra quyết định.
* Thông tin mức cao: bao gồm lời bài hát, nhịp điệu, tiết tấu... những đặc điểm này thường liên quan đến việc áp dụng kinh nghiệm của con người để xác định từng làn điệu cụ thể.
  + 1. Các ứng dụng của phân lớp làn điệu

Phân lớp thể loại nói chung và phân lớp theo làn điệu nói riêng có những ứng dụng tích cực trong những lĩnh vực sau:

* Tổng hợp có hệ thống các file dữ liệu âm nhạc.
* Hỗ trợ việc phân loại, tìm kiếm tên bài hát, làn điệu.
* Xây dựng các ứng dụng nhận diện tên bài hát, làn điệu.
  1. Một số phương pháp phân lớp thường dùng
     1. Máy học vector hỗ trợ(Support Vector Machines)
     2. Mạng nơ-ron nhân tạo(Neural Networks)
     3. Cây quyết định(Decision Trees)
     4. K-láng giềng gần nhất(K-Nearest Neighbors)

Giai đoạn 1: Phân cụm: chưa xác định cụ thể dữ liệu thuộc về làn điệu cụ thể nào, chỉ đưa về thể loại(chèo/quan họ) và nó thuộc vào.

Giai đoạn 2: Phân lớp: từ kết quả của *giai đoạn 1* nếu kết quả trả về chiếm phần trăm cao về phía cụm chèo thì dữ liệu sẽ được đưa vào nhận dạng làn điệu cụ thể trong lớp chèo, tương tự đối với làn điệu quan họ.

Phân chia dữ liệu huấn luyện và thử nghiệm



Dữ liệu thử nghiệm

Dữ liệu huấn luyện

***Huấn luyện***

***Thử nghiệm***

Cơ sở

dữ liệu

phân cụm

Tổng hợp kết quả phân cụm

Hình 1.1. Mô hình phân cụm chèo và dân ca quan họ

Hình 1.1 là mô hình phân cụm chèo và dân ca quan họ của **giai đoạn 1**. Đầu tiên dữ liệu âm thanh sẽ được trích chọn đặc trưng, loại bỏ những khoảng lặng, nhiễu, sau đó dữ liệu này sẽ được phân thành 5 phần: với 4 phần sẽ đưa vào để huấn luyện và 1 phần đưa vào thử nghiệm. Sau khi huấn luyện xong ta sẽ có một cơ sở dữ liệu liên quan đến hai cụm chèo và quan họ, công việc tiếp theo sẽ là thử nghiệm dữ liệu test bên trên với cơ sở dữ liệu này. Cuối cùng là việc tổng hợp kết quả để phục vụ cho việc phân lớp làn điệu cụ thể của **giai đoạn 2**.

Cơ sở dữ liệu

phân lớp chèo

Tổng hợp kết quả phân lớp

***Thử nghiệm***

***Thử nghiệm***

Cơ sở dữ liệu

phân lớp quan họ

***Huấn luyện***

***Huấn luyện***

Cụm dữ liệu chèo

Cụm dữ liệu quan họ

Đối chiếu với kết quả phân cụm

Dữ liệu thử nghiệm

Dữ liệu quan họ

Dữ liệu chèo

Dữ liệu huấn luyện

Hình 1.2. Mô hình phân lớp làn điệu chèo và dân ca quan họ

Hình 1.2 là mô hình phân lớp theo từng làn điệu chèo và dân ca quan họ cụ thể, sử dụng lại những kết quả có được thông qua giai đoạn 1, ta có được dữ liệu huấn luyện và thử nghiệm đã được trích chọn đặc trưng. Với dữ liệu huấn luyện ta đưa vào huấn luyện với mục đích tạo nên Cơ sở dữ liệu phân lớp theo từng làn điệu cụ thể, còn dữ liệu thử nghiệm thì ta so sánh với kết quả thu được ở giai đoạn một để phia dữ liệu thành hai phần, một phần để nhận diện về làn điệu chèo còn một phần để nhận diện về làn điệu quan họ. Cuối cùng tiến hành thử nghiệm và tổng hợp thành kết quả.

* 1. Tìm hiểu một số công trình nhận phân lớp âm nhạc đã có tại Việt Nam
     1. Phân loại nhạc Việt Nam theo thể loại dựa trên âm sắc và nhịp điệu tác giả: Phan Anh Cang, Nguyễn Thị Kim Khánh, Phan Thượng Cang.

1. Cơ sở dữ liệu

Nghiên cứu sử dụng nguồn dữ liệu cho huấn luyện và kiểm tra là bộ sưu tập nhạc GTZAN, gồm 10 thể loại nhạc. Dựa trên cấu trúc phân loại âm thanh của bộ sưu tập nhạc GTZAN, bốn thể loại nhạc Classical, Rock, Jazz, Pop được chọn ngẫu nhiên để minh họa cho hệ thống phân loại nhạc theo thể loại.

Dữ liệu được chia thành 2 tập: huấn luyện và kiểm tra. Mỗi file có độ dài 30s với tần số 22050Hz Mono 16-bit ở định dạng wav.

Tập file audio huấn luyện được sử dụng để huấn luyện cho bộ phân loại KNN để đưa ra các quyết định cho hệ thống phân loại nhạc trong khi tập file audio kiểm tra sẽ được sử dụng để đánh giá hiệu quả của phương pháp đề xuất.

1. Phương pháp nhận dạng

Hệ thống phân loại nhạc theo thể loại gồm 2 pha: rút trích đặc trưng và huấn luyện hoặc phân loại.

Hệ thống sử dụng phương pháp biến đổi Wavelet rời rạc(DWT) để rút trích đặc trưng về nhịp điệu. Phương pháp phân loại KNN được sử dụng để nhận dạng các thể loại nhạc. Quá trình huấn luyện bao gồm việc sử dụng các vector đặc trưng đã được gán nhãn thể loại để huấn luyện cho mô hình phân loại KNN. Từ đó, bộ phân loại sẽ gán nhãn thể loại cho các vector đặc trưng mới một cách tự động.

Vector đặc trưng:

* Âm sắc
* Nhịp điệu/tiết tấu
* Cao độ

Cơ sở dữ liệu nhạc

Cơ sở dữ liệu vector đặc trưng

Trích rút đặc trưng

Huấn luyện nhận dạng(Bộ phân loại KNN)

Vector đặc trưng:

* Âm sắc
* Nhịp điệu/tiết tấu
* Cao độ

Tín hiệu audio

Trích rút đặc trưng

Thể loại nhạc:

* Classical
* Rock
* Jazz
* Pop

Hình 1.3. Mô hình tổng quát hệ thống phân loại nhạc theo thể loại

Việc phân loại nhạc được thực hiện chủ yếu dựa vào 3 tập đặc trưng được rút trích từ tín hiệu audio như sau:

* Tập đặc trưng 1 (ĐT1): 19 đặc trưng về âm sắc.
* Tập đặc trưng 2 (ĐT2): 6 đặc trưng về nhịp điệu.
* Tập đặc trưng 3 (ĐT3): 5 đặc trưng về cao độ

1. Kết quả nhận dạng

Công trình tiến hành thử nghiệm trên một số giá trị tham số k(số láng giềng gần nhất). Trong đó, với giá trị k = 4, hệ thống cho kết quả phân loại tốt nhất. Việc phân loại dựa trên cả 3 tập đặc trưng thì thu được kết quả như bên dưới đây:

* Với k = 3, độ chính xác là 80,3%
* Với k = 4, độ chính xác là 83,5%
* Với k = 5, độ chính xác là 81,4%
* Với k = 6, độ chính xác 80,8%
* Với k = 7, độ chính xác 81,4%

CHƯƠNG 2. CÁC CÔNG CỤ SỬ DỤNG TRONG ĐỂ TÀI

1. 1. Bộ công cụ ALIZE
2. 1. 1. Nguồn gốc

ALIZE(ALIZÉ) là một thư viện mở hõ trợ cho việc nhận diện giọng nói. Mục đích của dự án này là cung cấp một bộ thư viện nền tảng cho phép bất kỳ ai cũng có thể phát triển các ứng dụng khác nhau trong lĩnh vực nhận dạng giọng nói như việc xác minh/xác định, phân lớp, v.v.

Dự án ALIZE được bắt đầu từ năm 2004, bởi đại học Avignon LIA thuộc tập đoàn ELISA với mục đích tạo ra một thư viện C++ mã nguồn mở để nhận dạng giọng nói. Tất cả mã nguồn của bộ công cụ này được phân phối thông qua giấy phép phần mềm nguồn mở(LPGL) và đã được thử nghiệm trên các nền tảng khác nhau bao gồm Window, Linux và Mac-OS.

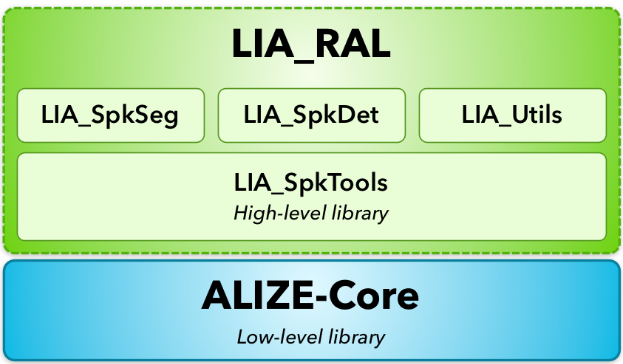
ALIZE là phần mềm miễn phí, có thể dowload tại địa chỉ: **http://alize.univ-avignon.fr/#download*.***

* + 1. Giới thiệu về ALIZE

ALIZE được phát triển với kiến trúc đa tầng dựa trên mô hình UML. Với thư viện lõi là ALIZE-Core(là một thư viện mức thấp bao gồm tất cả các công cụ cần thiết đề sử dụng mô hình hỗn hợp Gaussian – GMM), các chức năng mức cao dành cho việc nhận diện giọng nói được cài đặt trong gói LIA-RAL. Chính vì điều này hầu hết các hoạt động của người sử dụng đều thông qua LIA-RAL. Mã trong gói LIA-RAL được chia thành hai loại:

* Thứ nhất là thư viện mức cao(có tên LIA\_SpkTools), chứa các giải thuật huấn luyện và các hàm thống kê được áp dụng trên các véc tơ đặc trưng.
* Thứ hai là các bộ công cụ LIA\_SpkSeg, LIA\_SpkDet và LIA\_Utils.

Bộ thư viện ALIZE chỉ hỗ trợ các công cụ cần thiết để xây dựng một hệ thống xác minh/xác định, phân lớp,… không phải là một hệ thống hoàn chỉnh. Vì vậy, việc xây dựng nên hệ thống phân lớp đòi hỏi việc nghiên cứu tài liệu, tập tin cấu hình có sẵn của thư viện, xác định tính năng của mỗi chương trình con để từ đó giải quyết các bài toán cụ thể.

****

Hình 2.1. Thành phần của gói công cụ ALIZE

**Cụ thể về các bộ công cụ trong gói LIA\_RAL**:

* *LIA\_Utils:* công cụ cần thiết để xử lý tất cả các định dạng được sử dụng trong ALIZE: mô hình GMM, các thông số, ...
* *LIA\_SpkDet:* một tập hợp các công cụ để làm tất cả các nhiệm vụ được yêu cầu bởi một hệ thống xác thực sinh trắc học: huấn luyện mô hình (tiếng nói / từ), các thông số tiêu chuẩn hóa, chuẩn hóa điểm số, …
* *LIA\_SpkSeg:* một công cụ mới để phát hiện tiếng nói.
  + 1. Bộ công cụ SPro

SPro là một bộ công cụ cung cấp các lệnh thực thi các thuật toán trích chọn đặc trưng tiếng nói và trong các ứng dụng nhận dạng người nói.

SPro ban đầu được thiết kế để hiển thị giải phân tích quang phổ nhưng cũng cung cấp các kỹ thuật trích chọn đặc trưng cổ điển sử dụng trong ứng dụng tiếng nói. Có những lệnh chính sau đây:

* Năng lượng filter – bank
* Hệ số cepstral (filter-bank và dự đoán tuyến tính)
* Dự đoán tuyến tính dẫn xuất tiêu biểu (dự báo và phản ánh hệ số, tỷ lệ phạm vi và những dòng cặp phổ).

Các thư viện được viết trong ANSI C, cung cấp các chức năng sau:

* Tín hiệu đầu vào dạng sóng
* Xử lý tín hiệu ở mức thấp (FFT, phân tích LPC,..)
* Xử lý đặc trưng mức thấp ( CMS, chuẩn hóa sự khác biệt, …)

Thư viện không cung cấp các hàm cho trích chọn đặc trưng ở mức cao mà trực tiếp chuyển đổi một dạng sóng vào các đặc trưng, chủ yếu là do hàm như vậy sẽ đòi hỏi một số lượng lớn các đối số để được linh hoạt.

Spro được sử dụng trên nền Linux, SPARC/SunOS và HP-UX. Có thể dowload Spro phiên bản SPro4.0.1, dung lượng 1.3M tại địa chỉ đường link: **http://www.irisa.fr/metiss/guig/spro/download.html**

* + 1. Sử dụng SPro và ALIZE trong phân lớp làn điệu

**Tính kết quả**

**nhận dạng**

Huấn luyện mô hình đích (gmm)

**Train**

**Target**



Huấn luyện mô hình nền

Chuẩn hóa đặc trưng

Loại bỏ

khoảng lặng

Trích chọn đặc trưng

**NormFeat**

**TrainWorld**

**Compute**

**Test**

**Energy Detector**

**sfbcep**

**SPro**

**LIA\_SpkDet**

Hình 2.2. Sơ đồ sử dụng các công cụ của SPro và ALIZE trong phân lớp làn điệu

Hình 2.2 mô tả các bước và công cụ liên kết tại mỗi bước, dưới đây là trình bày các bước như sau:

1. Trích chọn đặc trưng

Bộ công cụ Spro sẽ đọc một file âm thanh đầu vào, xử lý nó và trích xuất các vector đặc trưng. Kết quả đầu ra sẽ là đầu vào cho hệ thống LIA\_SpkDet(ở trong Đồ án này em sử dụng bộ công cụ LIA\_SpkDet trong gói LIA\_RAL của ALIZE)

Các đặc trưng được trích chọn thực hiện bởi công cụ dựa trên phân tích filter –bank, sử dụng *sfbcep* trong *SPro*. Việc thực hiện dựa trên câu lệnh sau:

*sfbcep [option] inputfile outputfile*

Các tùy chọn(option) như sau:

*-F, --format=str*

Xác định sóng đầu vào của file. Định dạng hiển thị là ‘PCM16’, ‘wave’ hoặc ‘sphere’. Mặc định là: ‘PCM16’

*-f, --sample-rate=f*

Thiết lập sóng đầu vào tỉ lệ tần sô cho ‘PCM16’. Mặc định là 8kHz

*-x, --channel=n*

Thiết lập đặc trưng cho các kênh. Mặc định là 1

*-B, --swap*

Trao đổi các mẫu sóng đầu vào

*-I, --input-bufsize=n*

Thiết lập bộ đệm đầu vào kích cỡ tới n kbytes. Mặc định: 10MB

*-O, --output-bufsize=n*

Thiết lập bộ đệm đầu vào kích cỡ tới n kbytes. Mặc định: 10MB

*-k, --pre-emphasis=f*

Thiết lập hệ số tiền nhấn mạnh đến f. Mặc định: 0.95

*-l --length=f*

Thiết lập các phân tích dài khung tới *f ms.* Mặc định: 20.0 ms.

*-d, --shift=f*

Thiết lập khoảng thời gian giữa hai khung liên tiếp tới ms.

Mặc định là: 10ms

*-p, --num-ceps=n*

Thiết lập số lượng đầu ra hệ số Cepstral đến n. Mặc định là 12.

*-e, --energy*

Thêm năng lượng cho vector đặc trưng

*-w, --window=str*

Xác định các cửa sổ dạng sóng. Cửa sổ có sẵn là ‘Hamming’, ‘Hanning', `Blackman' hay không có gì. Mặc định là của số ‘Hamming’

*-n, --num-filters=n*

Thiết lập số kênh trong filter bank. Mặc định là: 24

*-a, --alpha=f*

Thiết lập ràng buộc tần số thấp đến f Hz. Mặc đinh: 0 Hz

*-u, --freq-max=f*

Thiết lập tần số giới hạn trên đến f Hz. Mặc định: Niquist.

*-b, --fft-length=n*

Thiết lập độ dài FFT tới n mẫu. Mặc đinh: 512

*-D, --delta*

Thêm các dẫn xuất đầu tiên đến các vector đặc trưng

*-A, --acceleration*

Thêm các dẫn xuất tự thứ hai với vector đặc trưng. Yêu cầu *`--delta'.*

*-v, --verbose*

Bật chế độ verbose

*-h, --help*

In thông báo và giúp các công cụ thoát

*-V, --version*

In thông tin phiên bản và thoát..

*Inputfile*: Đầu vào là đường dẫn đến file tín hiệu âm thanh

*Outputfile:* Đầu ra của file âm thanh

1. Loại bỏ khoảng lặng

Khi tất cả các vector đặc trưng được tính toán, một bước rất quan trọng là quyết định vector đó sử dụng được hay không. Một cách để phát hiện ra vấn đề đó là so sánh với những khung có khoảng lặng. Vì vậy, để chọn các khung tương ứng với tín hiệu, sử dụng một phương pháp tiếp cận dựa trên sự phân bố năng lượng của mỗi tín hiệu. Để chuẩn hóa các hệ số năng lượng, chạy lệnh *NormFeat* từ thư viện *LIA SpkDet*. Câu lệnh thực hiện như sau:

*NormFeat [option] --inputFeatureFileName inputfile*

*NormFeat* nhằm mục đích xử lý tín hiệu đầu vào liên quan đến đặc trưng bằng cách áp dụng bất kỳ tiêu chuẩn hóa

*inputfile:* là tên của file cần làm việc, là một danh sách, có phần mở rộng là *.lst*

*options:* là các tùy chọn.

Giá trị tùy chọn đượcmô tả ở bảng dưới đây:

Bảng 2.1: Mô tả tùy chọn sử dụng tiêu chuẩn hóa năng lượng

|  |  |  |
| --- | --- | --- |
| **Tùy chọn** | **Giá trị** | **Mô tả** |
| mode | .norm | Xác định các loại tiêu chuẩn hóa( Định mức đặc trưng bình thường là phân bố gaussion trên 1,0) |
| vectSize | 60 | Xác định kích thước của các đặc trưng |
| loadFeatureFileFormat  saveFeatureFileFormat | SPRO4 | Định dạng gọi, lưu các đặc trưng của file |
| loadFeatureFileExtension | .prm | Định dạng phần mở rộng để gọi đặc trưng của file (tương ứng với đầu ra của các bước trích chọn đặc trưng) |
| saveFeatureFileExtension | .enr.prm | Định dạng phần mở rộng để lưu đặc trưng của file |
| featureFilesPath | ./prm/ | Định dạng đường dẫn để lưu và gọi file đặc trưng |
| writeAllFeatures | True | Khi thiết lập đúng, tất cả các tính năng về tập tin ban đầu được ghi |

Sau đó, để chọn các khung với năng lượng cao nhất, sử dụng lệnh *EnergyDetector* của thư viện *LIA SpkDet*. Câu lệnh thực hiện như sau:

*EnergyDetector.exe [options] --inputFeatureFileName inputfile*

*EnergyDetector* nhằm mục đích phân tích thành phần năng lượng đặc trưng đầu vào bằng cách cung cấp nhãn năng lượng đầu ra ở mức cao. Đây là một kiểu phát hiện tiếng nói/ không nói

*Inputfile*: là tên của đặc trưng mà năng lượng đã được tiêu chuẩn hóa.

*options:* là các tùy chọn.

Giá trị tùy chọn đượcmô tả ở bảng dưới đây:

Bảng 2.2: Mô tả tùy chọn sử dụng phát hiện tín hiệu tiếng nói

|  |  |  |
| --- | --- | --- |
| **Tùy chọn** | **Giá trị** | **Mô tả** |
| labelOutputFrames | Speech | Nhãn ảnh hưởng đến khung được chọn |
| LabelFilesPath | ./lbl | Xác định đường dẫn lưu trữ file |
| saveLabelFileExtension | .lbl | Lưu trữ các file mở rộng |
| Alpha | 0.0 | Tỷ lệ khung hình được chọn trong gaussian trung tâm |
| nbTrainIt | 10 | Số bước lặp của EM để ước lượng năng lượng phân phối |
| baggedFrameProbabilityInit | 0.001 | Xác định tỷ lệ giữa các khung hình được lựa chọn và tổng số các khung được sử dụng để khởi tạo |

1. Chuẩn hóa đặc trưng

Các vector tham số được chuẩn hóa để phù hợp với trung bình bằng không và các thuộc tính không phù hợp. Giá trị trung bình và phương sai được sử dụng cho chuẩn hóa được tính bằng tập tin trên tổng số các khung sau khi xử lý loại bỏ khung.

Việc chuẩn hóa đặc trưng cũng giống như sử dụng chuẩn hóa năng lượng. Chỉ khác một điểm khi chạy lệnh *NormFeat* là gọi nhãn file. Vì vậy, để thực hiện các chuẩn hóa năng lượng, *NormFeat* chỉ sử dụng các khung với năng lượng cao nhất. Bảng dưới đề cập đến những thay đổi so với chuẩn hóa năng lượng:

Bảng 2.3: Mô tả tùy chọn chuẩn hóa đặc trưng

|  |  |  |
| --- | --- | --- |
| **Tùy chọn** | **Giá trị** | **Mô tả** |
| saveFeatureFileExtension | .norm.prm | Xác định định dạng để lưu file |
| labelSelectFrames | speech | Xác định nhãn để làm việc |
| LabelFilePath | ./prm | Đường dẫn để lưu file |

1. Huấn luyện mô hình nền

Mục đích của bước này là mô hình nền phổ quát (UBM) được tạo ra, sử dụng tất cả các dữ liệu mà chúng ta đã chuẩn hóa trước đó. Mô hình GMM được ước tính bằng cách sử dụng thuật toán EM (tối đa hóa kỳ vọng).

Huấn luyện mô hình nền, thực thi bởi lệnh *TrainWorld* của thư viện *LIA\_SpkDet.* Câu lệnh được thực hiện như sau:

*TrainWorld [options] --inputFeatureFileName inputfile*

*--outputWorldFilename worldfile*

*TrainWorld:* học mô hình GMM thông qua thuật toán EM.

*inputfile:* là tên của các chuẩn hóa đặc trưng để làm việc với nó và nó có thể là một danh sách mở rộng, có đuôi là .*lst*

*worldfile:* là tên file kết quả từ mô hình.

*options:* là các tùy chọn.

Giá trị tùy chọn đượcmô tả ở bảng dưới đây:

Bảng 2.4: Mô tả tùy chọn chính sử dụng huấn luyện mô hình từ

|  |  |  |
| --- | --- | --- |
| **Tùy chọn** | **Ví dụ** | **Mô tả** |
| distribType | GD | Xác định kiểu của thuộc tính |
| saveLabelFileExtension | .gmm | Định dạng để lưu thuộc tính |
| loadFeatureFileExtension | .norm.prm | Định dạng đầu vào của file |
| MixtureDistribCount | 20 | Xác định số thuộc tính gaussion trong hỗn hợp |
| mixtureFilePath | ./gmm | Đường dẫn để lưu thư mục file |
| baggedFrameProbabilityInit | 0.0005 | Xác định tỷ lệ giữa các khung được lựa chọn và tổng số các khung được sử dụng để khởi tạo. |
| baggedFrameProbability | 0.01 | Xác định tỷ lệ giữa các khung hình được lựa chọn và tổng số các khung được sử dụng để huấn luyện |
| nbTrainIt | 25 | Số lần lặp của EM liên quan đến baggedFrameProbabilityInit |
| nbTrainFinalIt | 4 | Số lần lặp của EM không có baggedFrameProbability |
| featureServerMask | 0-15, 17-33 | Được sử dụng để chọn một tập hợp các đặc trưng trong vector |
| vectSize | 33 | Xác định kích thước của các đặc trưng |
| NormalizeModel | true | Áp dụng N(0, 1) sắp xếp ở cuối quá trình huấn luyện |

1. Huấn luyện mô hình đích (Target model training)

Huấn luyện mô hình đích, chạy lệnh *TrainTarget* của gói *LIA\_SpkDet.* Sử dụng câu lệnh như sau:

*TrainTarget [options] --targetIdList inputfile*

*--inputWorldFilename worldfile*

*TrainTarget* nhằm mục đích huấn luyện đích người nói bằng cách làm phù hợp mô hình từ thông qua phương pháp MAP.

*Inputfile* là danh sách đầu, chính là danh sách các tên file đưa vào huấn luyện

*Worldfile: world -- debug false –verbose true*

*options:* là các tùy chọn.

Giá trị tùy chọn đượcmô tả ở bảng dưới đây:

Bảng 2.5: Mô tả tùy chọn trong huấn luyện mô hình đích

|  |  |  |
| --- | --- | --- |
| **Tùy chọn** | **Giá trị** | **Mô tả** |
| baggedFrameProbability | 0.1 | Xác định tỷ lệ giữa các khung được lựa chọn và tổng số các khung được sử dụng để thích ứng |
| nbTrainIt | 5 | Số lần lặp EM liên quan tới baggedFrameProbability |
| nbTrainFinalIt | 1 | Số lần lặp EM không liên quan tới baggedFrameProbability |
| targetIdList | Ndx | Danh sách đầu vào: mô hình trên cột đầu tiên, nhập vào tên tập tin đặc trưng về những người khác nhau |
| MAPAlgo | MAPOccDep | Xác định phương pháp thích ứng để sử dụng: MAPOccDep hoặc MAPConst |
| MAPRegFactor | 10 | Tham số được sử dụng bởi những kỹ thuật thích ứng MAPOccDep |

Phương thức MAP sử dụng là phương pháp tiếp cận MAPOCCDep: biến ngẫu nhiên để ước lượng được tính bằng sự kết hợp tuyến tính các giá trị trong mô hình nền với dữ liệu trên thuật toán EM. Phương pháp này sẽ đưa vào một xác suất hậu nghiệm cho mỗi Gaussion. Trọng số của sự kết hợp này được cung cấp bởi các tùy chọn MAPREGFactor r

1. Thử nghiệm

Mục tiêu của bước này là tính toán điểm số cho từng thử nghiệm của các đặc trưng vector để đưa ra một mô hình đích và mô hình nền (số điểm là xác suất tính được qua thử nghiệm từng file âm thanh). Để tính toán số điểm này, chỉ xem xét 10 thuộc tính Gaussian đầu tiên của mô hình. Câu lệnh được thực hiện như sau:

*ComputeTest [options] --ndxFileName inputfile*

*--worldModelName worldfile*

*--outputFile outputfile*

*ComputeTest:* đưa ra điểm số liên quan đến một phần thử nghiệm và mô hình đích.

*Inputfile:* là đưa ra danh sách các file thử nghiệm

*Worldfile:* là tên của mô hình nền

*Outputfile:* là kết quả ghi ra tập tin

*options:* là các tùy chọn.

Giá trị tùy chọn đượcmô tả ở bảng dưới đây:

Bảng 2.6: Mô tả tùy chọn tính điểm số của mô hình GMM

|  |  |  |
| --- | --- | --- |
| **Tùy chọn** | **Giá trị** | **Mô tả** |
| topDistribsCount | 10 | Số Gaussians sử dụng để tính toán điểm số |
| computeLLKWithTopDistribs | COMPLETE | Tính toán với phân phối đầu |
| Gender | M | Giới tính của tập tin ndx |
| loadMixtureFileExtension | .gmm | Xác định phần mở rộng để gọi mô hình |
| loadFeatureFileExtension | .norm.prm | Xác định phần mở rộng để gọi đặc trưng |
| mixtureFilesPath | ./gmm | Xác định đường dẫn để gọi hỗn hợp |
| outputFile | test1.res | Kết quả điểm số tập tin |

CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG THỬ NGHIỆM NHẬN DẠNG LÀN ĐIỆU CHÈO VÀ DÂN CA QUAN HỌ

1. 1. Sơ đồ tổng quan quá trình xây dựng hệ thống nhận dạng làn điệu chèo và dân ca quan họ

|  |
| --- |
| Chia dữ liệu thành 5 phần bao gồm 4 phần để train và 1 phần để test    Chuẩn bị dữ liệu      Thực hiện train và phân dữ liệu test đầu vào thành 2 lớp: chèo và quan họ      Đưa những file test có kết quả phân lớp theo thể loại đúng vào nhận diện theo làn điêụ cụ thể    Tổng hợp kết quả |

Hình 3.1. Hệ thống phân lớp làn điệu chèo và dân ca quan họ

Hình 3.1 là các bước thực hiện của hệ thống phân lớp làn điệu được thử nghiệm trong đề tài. Bước đầu tiên là việc chuẩn bị dữ liệu, dữ liệu này là các file âm thanh có định dạng là .wav, mono, với tần số 16000Hz, được cắt bỏ phần nhạc dạo bằng phần mềm Wavepad Sound Editor. Danh sách các làn điệu sẽ được trình bày cụ thể ở mục 3.2 của chương này. Sau khi chạy chương trình thử nghiệm trên bộ công cụ ALIZE sẽ cho ra số(Score) của file âm thanh đưa vào test với danh sách các làn điệu đã được train. File âm thanh này sẽ được xếp vào làn điệu nào được đánh điểm cao nhất.

* 1. Chuẩn bị dữ liệu

Chương trình được thử nghiệm trên 25 làn điệu chèo và 25 làn điệu quan họ. Thông tin của toàn bộ các làn điệu được mô tả trong bảng dưới đây:

**Bảng 3.1: Thông tin về 25 làn điệu chèo**

|  |  |  |
| --- | --- | --- |
| STT | Tên làn điệu | Tên viết tắt |
| 01 | Bình thảo | CH\_BT |
| 02 | Cấm giá | CH\_CG |
| 03 | Chinh phụ | CH\_CP |
| 04 | Chức cẩm hồi văn | CH\_CCHV |
| 05 | Con nhện giăng mùng | CH\_CNGM |
| 06 | Đò đưa | CH\_DD |
| 07 | Đào liễu | CH\_DL |
| 08 | Đào lý một cành | CH\_DLMC |
| 09 | Duyên phận phải chiều | CH\_DPPC |
| 10 | Đường trường bắn thước | CH\_DTBT |
| 11 | Đường trường tiếng đàn | CH\_DTTK |
| 12 | Đường trường thu không | CH\_DTTK |
| 13 | Đường trường trên non | CH\_DTTN |
| 14 | Đường trường vị thủy | CH\_DTVT |
| 15 | Du xuân | CH\_DXN |
| 16 | Dương xuân | CH\_DX |
| 17 | Lới lơ | CH\_LL |
| 18 | Luyện năm cung | CH\_LNC |
| 19 | Nón thúng quai thao | CH\_NTQT |
| 20 | Quá giang | CH\_QG |
| 21 | Quạt màn | CH\_QM |
| 22 | Quân tử vu dịch | CH\_QTVD |
| 23 | Sắp qua cầu | CH\_SQC |
| 24 | Tò vò | CH\_TV |
| 25 | Xẩm xoan | CH\_XX |

**Bảng 3.2: Thông tin về 25 làn điệu quan họ**

|  |  |  |
| --- | --- | --- |
| STT | Tên làn điệu | Tên viết tắt |
| 01 | Ba quan | QH\_BQ |
| 02 | Bèo dạt mây trôi | QH\_BDMT |
| 03 | Buôn bấc buôn dầu | QH\_BBBD |
| 04 | Chuông vàng gác cửa tam quan | QH\_CVGCTQ |
| 05 | Còn duyên | QH\_CD |
| 06 | Cổ tay vừa trắng vừa tròn | QH\_CTVTVT |
| 07 | Đôi bên bác mẹ cùng già | QH\_DBBMCG |
| 08 | Đối ca sông cầu | QH\_DCSC |
| 09 | Gió đưa cây cải | QH\_GDCC |
| 10 | Giăng thanh gió mát | QH\_GTGM |
| 11 | Lý giao duyên | QH\_LGD |
| 12 | Lòng vẫn đợi chờ | QH\_LVDC |
| 13 | Nguyệt gác mái đình | QH\_NGMD |
| 14 | Năm liệu bảy lo | QH\_NLBL |
| 15 | Nhớ mãi khôn nguôi | QH\_NMKN |
| 16 | Người ơi đến hẹn lại về | QH\_NODHLV |
| 17 | Se chỉ luồn kim | QH\_SCLK |
| 18 | Súc miệng ấm đồng | QH\_SMAD |
| 19 | Tưởng đến gần xa | QH\_TDGX |
| 20 | Tìm em trong chiều hội Lim | QH\_TETCHL |
| 21 | Thân lươn bao quản lấm đầu | QH\_TLBQLD |
| 22 | Tương phùng tương ngộ | QH\_TPTN |
| 23 | Trên rừng ba mươi sáu thứ chim | QH\_TRBMSTC |
| 24 | Tay tiên chuốc chén rượu đào | QH\_TTCCRD |
| 25 | Vui bốn mùa | QH\_VBM |

File nhạc sau khi được tải về từ trên mạng internet sẽ cắt và chuyển về định dạng file Wav. Với việc tiến hành phân lớp theo thể loại(chèo hay quan họ) thì sẽ được huấn luyện với 25\*16 = 400 file âm thanh, sau đó khi tiến hành phân lớp theo làn điệu thì mỗi làn điệu sẽ được huấn luyện với 16 file âm thanh.

Kết quả đã tạo được cơ sở dữ liệu âm nhạc phục vụ cho quá trình phân lớp làn điệu chèo và dân ca quan họ. Cơ sở dữ liệu này được đặt trong thư mục **wav**: gồm có 1000 file âm thanh với tổng kích thước là 1,8 GB.

Với mỗi làn điệu, định dạng file được được lưu dựa theo tên viết tắt được đề cập trong hai bảng liệt kê danh sách làn điệu chèo(bảng 3.1) và làn điệu dân ca quan họ(bảng 3.2) phía trên. Định dạng file được tổ chức như sau:

CH/QH\_Chữ cái bắt đầu của từng chữ trong tên làn điệu-Thứ tự trong của file gốc trong danh sách các file của một làn điệu-Vị trí của phần được cắt ra trong file gốc.

Trong đó:

* CH: Ký hiệu là file chèo
* QH: Ký hiệu là file dân ca quan họ

Ví dụ: làn điệu Bình Thảo có các file mp3 gốc là:

* 1\_BinhThao-ThuHuyen\_ThanhTam.mp3
* 2\_BinhThao-MinhToan.mp3
* 3\_BinhThao-DoanThanhBinh.mp3

Từ file 1\_BinhThao-ThuHuyen\_ThanhTam.mp3, ta cắt ra được 3 file wav, thì định dạng tên của từng file sẽ là: CH\_BT-01-01, CH\_BT-01-02, CH\_BT-01-03. Tương tự với các file khác ta sẽ được là CH\_BT-02-01, CH\_BT-02-02,…

Dữ liệu được mô tả ở trên được chia thành 2 tập:

* **Tập dữ liệu dành để huấn luyện:** gồm dữ liệu của 50 làn điệu(25 làn điệu chèo, 25 làn điệu quan họ), mỗi làn điệu sẽ có 16 file được đưa vào huấn luyện. tổng cộng là 800 file
* **Tập dữ liệu để Test:** Lấy 4 file trong mỗi làn điệu để làm dữ liệu test

Các file được dùng để huấn luyện và test được luân phiên đổi chỗ cho nhau. Ví dụ 20 file của một làn điệu sẽ được chia ra làm 5 phần: phần 1, phần 2, phần 3, phần 4, phần 5. Đầu tiên sẽ lấy phần 1 làm test, còn các phần 2, 3, 4, 5 sẽ đưa vào huấn luyện, sau khi thu được kết quả thì lại lấy phần 2 làm test, các phần 1, 3, 4, 5 đưa vào huấn luyện,… Thực hiện liên tục cho đến khi tất cả các file đều đã được test, thì tiến hành tổng hợp kết quả.

* 1. Phân lớp làn điệu sử dụng ALIZE

Dù là việc phân cụm theo thể loại hay phân lớp thành từng làn điệu cụ thể thì cũng cân thực hiện các bước được mô tả bên dưới đây:

Tạo thư mục làm việc

Tính tham số MFCC

Dò tìm năng lượng

Phát hiện tiếng nói trong tín hiệu

Chuẩn hóa các tham số của tín hiệu

Chuẩn hóa TrainWorldInit

Chuẩn hóa TrainWorldFinal

Huấn luyện GMM cho từng thể loại/làn điệu

Tính điểm số của từng file đưa vào thử nghiệm

Hình 3.1. Các bước xây dựng hệ phân lớp sử dụng SPro & ALIZE

*Sau đây là hướng dẫn chi tiết các bước thực hiện lệnh chạy của chương trình:*

* + 1. Bước 1: Tạo thư mục làm việc

Tạo thư mục Alize chứa các file và thư mục làm việc. Bên trong thư mục này ta tạo các thư mục sau:

* cfg: chứa những file config.
* gmm, gmm\_ch, gmm\_qh: chứa những file huấn luyên GMM cho từng thể loại và từng làn điệu.
* lbl: chứa những file đánh dấu đoạn có tiếng nói trong dữ liệu đầu vào(bao gồm cả dữ liệu huấn luyện và thử nghiệm).
* lst, lst\_ch, lst\_qh: chứa danh sách các file đầu vào.
* ndx. ndx\_ch, ndx\_qh: chứa tên của những file sẽ được đưa vào huấn luyện GMM.
* prm: chứa những file đã được trích chọn đặc trưng.
* test, test\_ch. test\_qh: chứa những file sẽ đưa vào thử nghiệm.
* res. res\_ch, res\_qh: chứa kết quả của những file thử nghiệm.
* rpt: nơi lưu file báo cáo của toàn bộ quá trình thử nghiệm.
* wav: nơi lưu trữ toàn bộ những file âm thanh đầu vào.
  + 1. Bước 2: Tính tham số MFCC

Tạo ra các vector đặc trưng từ các tín hiệu tiếng nói, sử dụng công cụ *sfbcep* để tính các tham số MFCC. Điều kiện cần thiết để tạo được tham số MFCC là phải chuẩn hóa các file .wav thành các file .prm. Tạo thư mục có tên là “prm“ để chương trình chạy ghi lại các file .prm

* Đầu vào: file tiếng nói có định dạng \*.wav
* Đầu ra: file prm có chứa tham số MFCC

Sử dụng câu lệnh:

*./sfbcep.exe -F PCM16 -f 16000 -p 19 -e -D -A wav/CH\_BT-01-01.wav prm/CH\_BT-01-01.prm*

Ý nghĩa của câu lệnh như sau:

*-FPCM16*: sử dụng định dạng hiển thị sóng đầu vào của file là : PCM16

*-f 16000*: sử dụng tần số cho sóng đầu vào là: 16000Hz

*-p 19*: có hệ số Cepstral là 19

*-e*: sử dụng thêm năng lượng cho vector đặc trưng

*-D*: Thêm các dẫn xuất đầu tiên đến các vector đặc trưng

*-A*: Thêm các dẫn xuất thứ hai với vector đặc trưng

🡺 Kết quả : sinh ra file *CH\_BT-01-01.prm*

* + 1. Bước 3: Dò tìm năng lượng
* Đầu vào: là những file .prm tạo được từ bước 2
* Đầu ra: file *.enr.prm* được lưu trong thư mục *prm*

Sử dụng câu lệnh:

*./NormFeat.exe --config ./cfg/NormFeat\_energy.cfg --inputFeatureFilename “CH\_BT-01-01”*

🡺 Kết quả : sinh ra file *CH\_BT-01-01.enr.prm*

* + 1. Bước 4: Phát hiện tiếng nói trong tín hiệu

Mục đích để phát hiện những đoạn có tiếng nói

* Đầu vào: là những file .prm được tạo ra từ bước 2
* Đầu ra: file *.lbl* được lưu trong thư mục *lbl*

Sử dụng câu lệnh:

*./EnergyDetector.exe --config ./cfg/EnergyDetector.cfg --inputFeatureFilename “CH\_BT-01-01”*

🡺 Kết quả : sinh ra file *CH\_BT-01-01.lbl*

Nội dung của file *CH\_BT-01-01.lbl*:

*0 0 speech*

*0.18 0.18 speech*

*0.2 0.21 speech*

*0.23 0.25 speech*

*…*

*43.87 43.88 speech*

*43.97 43.97 speech*

*44.02 44.02 speech*

* + 1. Bước 5: Chuẩn hóa các tham số của tín hiệu
* Đầu vào: là những file .prm tạo được từ bước 2
* Đầu ra: file *.norm.prm* được lưu trong thư mục prm

Sử dụng câu lệnh:

*./NormFeat.exe --config ./cfg/NormFeat.cfg --inputFeatureFilename “CH\_BT-01-01”*

🡺 Kết quả : sinh ra file *CH\_BT-01-01.norm.prm*

* + 1. Bước 6: Tạo sanh sách file wav
* Đầu vào: là những file trong thư mục wav
* Đầu ra: tạo ra trong thư mục lst 2 file: file all.lst chứa đầy đủ thông tin của file âm thanh, all1.lst chỉ chứa tên của file âm thanh

Ví dụ: Nội dung của file all.lst

*CH\_BT-01-01.wav*

*CH\_BT-01-02.wav*

*CH\_BT-01-03.wav*

*CH\_BT-02\_01.wav*

*……………*

*QH\_VBM-05-04.wav*

*QH\_VBM-06-02.wav*

*QH\_VBM-06\_03.wav*

*QH\_VBM -06\_04.wav*

Nội dung của file all1.lst:

*CH\_BT-01-01*

*CH\_BT-01-02*

*CH\_BT-01-03*

*CH\_BT-02\_01*

*……………*

*QH\_VBM-05-04*

*QH\_VBM-06-02*

* + 1. Bước 7: Phân chia dữ liệu thành hai phần huấn luyện và thử nghiệm
* Đầu vào: là file all1.lst tạo được ở Bước 6.
* Đầu ra: là những file có định dạng TEN\_THE\_LOAItrain.lst và TEN\_THE\_LOAItest.lst trong thư mục lst với mục đích thực hiện phân lớp theo thể loại hoặc TEN\_LAN\_DIEUtrain.lst và TEN\_LAN\_DIEUtest.lst với mục đích thực hiện phân lớp theo làn điệu.

Ví dụ: File *CHtrain.lst* trong thư mục lst sẽ có nội dung như sau:

*CH\_BT-02-02*

*CH\_BT-02-03*

*CH\_BT-03-01*

*CH\_BT-03-02*

*CH\_BT-03-03*

*…*

*CH\_XX-10-01*

*CH\_XX-10-02*

* + 1. Bước 8: Tạo file World.lst và World.lenght
* ***World.lst:*** nội dung của file này là chỉ đường dẫn đến tất cả các danh sách file âm thanh được dùng để huấn luyện (chính là file *TEN\_THE\_LOAItrain.lst* hoặc TEN\_LAN\_DIEUtrain.lst được tạo ra ở Bước 7). Mỗi tên file được lưu trên một dòng.

Ví dụ: Nội dung của file World.lst trong thư mục lst

*D:\Alize\lst\CHtrain.lst*

*D:\Alize\lst\QHtrain.lst*

* ***world.weight***: file trọng số, với việc phân theo 2 lớp(chèo hoặc quan họ) nên trọng số sẽ là 0.5.

Ví dụ: file World.weight trong thư mục lst

*0.5*

*0.5*

* + 1. Bước 9: Chuẩn hóa TrainWorldInit
* Đầu vào: file *world.lst* và file *world.weight*.
* Đầu ra: file *worl\_init.gmm, world\_initinit.gmm.*

*Sử dụng câu lệnh:*

*./TrainWorld.exe --config ./cfg/TrainWorldInit.cfg --inputStreamList ./lst/world.lst --weightStreamList ./lst/world.weight --outputWorldFilename world\_init --debug false --verbose true*

* + 1. Bước 10: Chuẩn hóa TrainWorldFinal
* Đầu vào: file *world.lst* và file *world.weight*
* Đầu ra: file *world.gmm*

Sử dụng câu lệnh:

*./TrainWorld.exe --config ./cfg/TrainWorldFinal.cfg --inputStreamList ./lst/world.lst --weightStreamList ./lst/world.weight --outputWorldFilename world --inputWorldFilename world\_init --debug false --verbose true*

* + 1. Bước 11: Tạo file chứa danh sách các file được huấn luyện
* Đầu vào: là những file có định dạng TEN\_THE\_LOAItrain.lst trong thư mục lst đối với việc phân lớp theo thể loại hoặc TEN\_LAN\_DIEUtrain.lst trong thư mục lst\_ch và lst\_qh đối với việc phân lớp theo làn điệu cụ thể.
* Đầu ra: là những file có định dạng TEN\_THE\_LOAI.ndx trong thư mục ndx với việc phân lớp theo thể loại hoặc TEN\_LAN\_DIEU.ndx trong thư mục ndx\_ch và ndx\_qh đối với việc phân lớp theo làn điệu cụ thể.

Ví dụ: file ch.ndx sẽ cấu trúc như sau:

*CH\_gmm CH\_BT-01-01 CH\_BT-01-02 CH\_BT-01-03 CH\_BT-02-01 CH\_BT-03-03 CH\_BT-04-02 CH\_BT-04-03 CH\_BT-05-02 CH\_BT-05-03 CH\_BT-06-01 CH\_BT-06-02 CH\_BT-07-01 CH\_BT-07-02 CH\_BT-08-01 CH\_BT-08-02 CH\_BT-08-03 CH\_CCHV-03-04 CH\_CCHV-04-02 CH\_CCHV-04-03 CH\_CCHV-05-02 CH\_CCHV-06-04*

*........* *CH\_XX-07-01 CH\_XX-07-02 CH\_XX-08-01 CH\_XX-08-02 CH\_XX-09-01 CH\_XX-09-02 CH\_XX-10-01 CH\_XX-10-02*

* + 1. Bước 12: Huấn luyện GMM
* Đầu vào: file *TEN\_THE\_LOAI.ndx* đối với việc phân lớp theo thể loại hoặc là file TEN\_LAN\_DIEU.ndx đối với việc phân lớp theo làn điệu cụ thể.
* Đầu ra: file *TEN\_THE\_LOAI\_gmm.gmm* trong thư mục gmm đối với việc phân lớp theo thể loại hoặc TEN\_LAN\_DIEU\_gmm.gmm trong thư mục gmm\_ch và gmm\_qh đối với việc phân lớp theo làn điệu cụ thể.

Câu lệnh thực hiện:

*./TrainTarget.exe --config ./cfg/target.cfg --targetIdList ./ndx/ch.ndx --inputWorldFilename world --debug false --verbose true*

🡺 Kết quả : sinh ra file *CH\_gmm*

* + 1. Bước 13: Tạo file thử nghiệm

Công việc này sẽ thực hiện khác nhau tùy thuộc vào từng giai đoạn là phân lớp theo thể loại hoặc phân lớp theo làn điệu cụ thể.

1. Phân lớp theo thể loại

* Đầu vào: những file có định dạng TEN\_THE\_LOAItest.lst trong thư mục lst và những file trong thư mục ndx.
* Đầu ra: là những file có định dạng TEN\_FILE\_AM\_THANH.ndx lưu trong thư mục test.

Ví dụ: nội dung của file CH\_BT-01-01.ndx là:

*CH\_BT-01-01 ch\_gmm qh\_gmm*

1. Phân lớp theo làn điệu cụ thể

* Đầu vào: là file report.txt tổng hợp kết quả từ việc phân lớp theo thể loại, những file kết quả trong thư mục res và những file trong 2 thư mục ndx\_ch và ndx\_qh.
* Đầu ra: là những file có định dạng TEN\_FILE\_AM\_THANH.ndx lưu trong thư mục test\_ch hoặc test\_qh.

Ví dụ: nội dung của file CH-BT-01-01.ndx lưu trong thư mục test\_ch là:

*CH\_BT-01-01 ch\_bt\_gmm ch\_cchv\_gmm ch\_cg\_gmm ch\_cngm\_gmm ch\_cp\_gmm ch\_dd\_gmm ch\_dl\_gmm ch\_dlmc\_gmm ch\_dppc\_gmm ch\_dtbt\_gmm ch\_dttd\_gmm ch\_dttk\_gmm ch\_dttn\_gmm ch\_dx\_gmm ch\_dxn\_gmm ch\_ll\_gmm ch\_lnc\_gmm ch\_ntqt\_gmm ch\_qg\_gmm ch\_qm\_gmm ch\_qtvd\_gmm ch\_slc\_gmm ch\_sqc\_gmm ch\_tv\_gmm ch\_xx\_gmm*

* + 1. Bước 14: Tiến hành thử nghiệm
* Đầu vào: là những file trong thư mục test đối với phân lớp theo thể loại, là những file trong thư mục test\_ch và test\_qh đối với việc phân lớp theo làn điệu cụ thể.
* Đầu ra : ra những file có định dạng FILE\_AM\_THANH.res lưu trong thư mục res đối với phân lớp theo thể loại hoặc res\_ch và res\_qh đối với việc phân lớp theo làn điệu.

Câu lệnh thực hiện:

*ComputeTest --config cfg/target\_seg\_male.cfg --ndxFilename ./test/CH\_BT-01-01.ndx --inputWorldFilename world --outputFilename ./res/CH\_BT-01-01.res*

🡺 Kết quả : sinh ra file *CH\_BT-01-01.res trong thư mục res*

Ví dụ: file kết quả *CH\_BT-01-01.res* trong thư mục res như sau:

*M ch\_gmm 1 CH\_BT-01-01 0.0751762*

*M qh\_gmm 0 CH\_BT-01-01 -0.103114*

Mỗi một file âm thanh sẽ ghi ra điểm số (score) tương ứng với N thể loại hoặc làn điệu. Điểm số của thể loại/làn điệu nào lớn nhất trong N thể loại/làn điệu chính là thể loại/làn điệu mà hệ thống cho ra kết quả cần tìm.

Chương 4. KẾT QUẢ THỬ NGHIỆM

Chương 5. KẾT LUẬN



5.1. Những vấn đề đã giải quyết trong Đồ án

Trong luận văn, tác giả đã sử dụng 2 phần mềm mã nguồn mở để áp dụng cho bài toán nhận dạng là công cụ Alize và Sphinx. 2 phần mềm này đã được thử nghiệm trên một số nước: Pháp, Anh, Thụy Điển, Đức, Thái …. Nhưng với đặc thù tiếng nói tiếng Việt, tác giả cũng gặp phải 1 số vấn đề khó khăn: gói công cụ Alize không dễ sử dụng do nó có rất ít tài liệu hỗ trợ. Hơn nữa, có những phần tài liệu kèm theo không phản ánh được việc cài đặt hiện tại.

Tuy chưa có đóng góp quan trọng nào vào trong hai công cụ này nhưng tôi đã mất rất nhiều thời gian đọc và tìm hiểu để có thể chạy hoàn chỉnh ứng dụng trên môi trường Linux.

Với đề tài: “ ***Nhận dạng người nói phụ thuộc vào từ khóa tiếng Việt***” tác giả đã giải quyết một số vấn đề sau:

* Tìm hiểu tổng quan bài toán nhận dạng tiếng nói nói chung và nhận dạng người nói phụ thuộc từ khóa nói riêng.
* Nghiên cứu các công cụ thích hợp để lựa chọn, áp dụng vào bài toán cụ thể và giới thiệu các chức năng, cách cài đặt của gói công cụ Alize và Sphinx
* Xây dựng được kho dữ liệu là các từ khóa tiếng Việt các số từ 0 - > 9 với 48 người nói ở các độ tuổi khác nhau ở phương ngữ Bắc của Việt Nam
* Phát triển và kết hợp 2 chương trình nhận dạng với bộ dữ liệu thử nghiệm người nói đạt độ chính xác cao, tuy thu âm với số lượng người chưa nhiều, chủ yếu là ở các tỉnh, huyện lân cận nhau Hưng Yên, Hải Dương, Bắc Ninh, Hà Nam với giọng nói chưa khác biệt nhau nhiều lắm nhưng với bộ từ khóa đủ lớn tỷ lệ nhận dạng đạt độ chính xác tuyệt đối 100%. Đây cũng là một bước nghiên cứu đóng góp vào phát triển thực tiễn cho lĩnh vực từ động nhận dạng người nói với từ khóa tiếng Việt, ứng dụng rất lớn trong thực tế để đăng nhập vào các tài khoản cá nhân thay password hay mã pin … như hiện nay.
  1. Hướng phát triển của đề tài

Với những kết quả đã đạt được, hướng phát triển của đề tài là:

- Tiếp tục xây dựng kho dữ liệu người nói đa dạng hơn: bao gồm số lượng người nói lớn và các phương ngữ khác nhau để đưa vào thử nghiệm trên ứng dụng.

- Phát triển ứng dụng người nói với bộ từ khóa ngắn làm sao để đem lại kết quả nhận dạng cao hơn.

- Kết hợp với ứng dụng người nói không phụ thuộc từ khóa và văn bản để thêm tính đa dạng của ứng dụng trong thực tế.

TÀI LIỆU THAM KHẢO

[1] Bonastre J.-F., Scheffer N., Fredouille C., Matrouf D. (2004), “Nist’04 Speaker Recognition Evaluation Campaign: New Lia Speaker Detection Plateform Based On Alize Toolkit”, *2004 NIST SRE 04 Workshop: speaker detection evaluation campaign*, Toledo, Spain.

[2] Bonastre J-F, Scheffer N., Matrouf D., Fredouille C., Larcher A., Preti A., Pouchoulin G., Evans N., Fauve B. and Mason J. (2008), “ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition”, *Proc. Odyssey: the Speaker and Language Recognition Workshop*.

[3] Dương Tử Cường (2003), *Xử lý tín hiệu số*, Nxb Quân đội nhân dân, Hà

Nội

[4] Đinh Lê Thành, Nguyễn Quốc Linh, Trịnh Văn Loan, “Định danh người nói Tiếng việt sử dụng mô hình hỗn hợp Gaussian “, *Hội nghị khoa học lần thứ 20 – ĐH Bách khoa Hà Nội,* tr. 40-45.

[4] Jean-Franc¸ois Bonastre, Fr´ed´eric Wils (2005), “ALIZE, A FREE TOOLKIT FOR SPEAKER RECOGNITION”, , pp. I 737 - I 740

[5] Ngô Minh Dũng, Đặng Văn Chuyết (2007), “ Xây dựng và khảo sát độ dài từ khóa trong nhận dạng người nói phụ thuộc vào từ khóa tiếng Việt theo mô hình Markov ẩn”, *Tạp chí BCVT & CNTT* ), tr. 93-99.

[6] Ngô Minh Dũng, Đặng Văn Chuyết (2004), “Khảo sát tính ổn định của một số đặc trưng ngữ âm trong nhận dạng người nói”, *Tạp chí BCVT & CNTT,* (kỳ 3 10/2004), tr. 12-18.

[7] Viện công nghệ thông tin Việt Nam (2004), *Tổng hợp và nhận dạng tiếng* *Việt*, Đề tài nghiên cứu cấp nhà nước KC01- 03D, Hà Nội

Website

8. <http://arxiv.org/ftp/arxiv/papers/1205/1205.3316.pdf>

9. <http://cmusphinx.sourceforge.net/wiki/tutorialpocketsphinx>

10. <http://cmusphinx.sourceforge.net/>

11. <http://en.wikipedia.org/wiki/Speech_recognition>

12. <http://home.iitb.ac.in/~jigar23/documents/Sphinx_Guideline_v1.pdf>

13. <http://mistral.univ-avignon.fr/index_en.html>

14. <http://ronaldramdhan.wordpress.com/2010/03/11/sphinxtrain/>

15. <http://www.irisa.fr/metiss/guig/spro/spro-4.0.1/spro_1.html#SEC1>

16. <http://www.speech.cs.cmu.edu/sphinx/tutorial.html>

17. <http://www.fon.hum.uva.nl/praat/>

PHỤ LỤC

1. Mô hình hỗn hợp Gauss

B. Ba bài toán cơ bản của HMM

B.1. Bài toán thứ nhất: Đánh giá xác suất

Cho dãy quan sát O = (o1o2…oT) và mô hình Markov ẩn λ = (A, B, π). Hãy tính P(O|λ) là xác suất của dãy quan sát theo mô hình.

Đây là bài toán đánh giá, tức là cho trước mô hình và dãy quan sát, ta phải tính toán xác suất mô hình tạo ra dãy quan sát đó như thế nào? Có thể xem đây là một kiểu đánh giá xem mô hình được cho có tốt với dãy quan sát đó hay không. Trong trường hợp ta phải chọn một trong số nhiều mô hình thì giải pháp cho bài toán 1 sẽ cho ta sự lựa chọn được mô hình thích hợp nhất đối với dãy quan sát.

Để giải quyết bài toán 1 ta sử dụng thủ tục tiến và thủ tục lùi được mô tả như sau:

Thủ tục tiến (Forward Procedure)

Xét biến tiến αt(i) được định nghĩa như sau:

αt(i) = P(o1o2…ot , qt=i|λ) (2.10)

Đó là xác suất của dãy quan sát bộ phận o1o2…ot và trạng thái i đạt được tại thời điểm t.

Ta có thể dùng phương pháp quy nạp để tính αt(i) như sau:

Bước 1. Khởi tạo:

α1(i) = πibi(o1) 1 ≤ i ≤ N (2.11)

Bước 2. Quy nạp:

 1 ≤ t ≤ T-1, 1 ≤ j ≤ N (2.12)

Bước 3. Kết thúc:

 (2.13)

Thủ tục lùi (Backward Procedure)

Tương tự ta có thể xét biến lùi βt(i) được định nghĩa như sau:

βt(i) = P(ot+1, ot+2 , oT , qt=i|λ) (2.14)

βt(i) là xác suất của dãy quan sát bộ phận từ thời điểm t+1 đến thời điểm cuối, căn cứ vào trạng thái i tại thời điểm t và mô hình λ. Có thể tính βt(i) bằng phương pháp quy nạp như sau:

**Bước 1. Khởi đầu:**

**βT(i) = 1 1 ≤ i ≤ N (2.15)**

**Bước 2. Quy nạp:**

**** (2.16)

**t = T – 1, T – 2, …, 1 1 ≤ i ≤ N**

**Bước 3. Kết thúc:**

**** (2.17)

Ở các phần sau ta sẽ thấy cách áp dụng các thủ tục tiến và lùi trong việc giải quyết vấn đề 3 của mô hình Markov ẩn.

B.2. Thuật toán thứ hai: Tìm dãy trạng thái tối ưu

Cho dãy quan sát O=(o1o2…oT) và mô hình λ. Làm thế nào lựa chọn được 1 dãy trạng thái q=(q1q2…qT) là tối ưu theo một nghĩa nào đó.

Đây là bài toán trong đó ta cố gắng tháo gỡ phần “ẩn” của mô hình để tìm ra dãy trạng thái “đúng” ẩn bên dưới. Không giống như bài toán 1 với kết quả cho ra là chính xác, bài toán 2 có nhiều cách giải quyết, tức là có nhiều cách để tìm ra dãy trạng thái “tối ưu” ứng với dãy quan sát cho trước. Khó khăn nằm ở chỗ định nghĩa dãy trạng thái thế nào là tối ưu – đó là vì có nhiều tiêu chuẩn tối ưu. Một tiêu chuẩn tối ưu là làm cực đại P(q|O,λ) cũng là làm cực đại P(q,O|λ). Phương pháp quy hoạch động để tìm chuỗi trạng thái đơn tốt nhất chính là giải thuật Viterbi sau đây.

Thuật toán Viterbi

Để tìm dãy trạng thái đơn tốt nhất q = (q1q2…qT) đối với dãy quan sát O = (o1o2…oT), ta định nghĩa đại lượng sau:



*δ­t(i)* là xác suất lớn nhất dọc theo một đường đi đơn tại thời điểm t của t quan sát đầu tiên trong đó có quan sát cuối cùng là ở trạng thái i.

Bằng quy nạp ta có:



Ta sử dụng mảng *ψt(j)* để giữa lại các đối số làm cực đại biểu thức trên ứng với mỗi t và j.

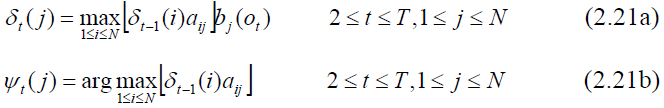
Bước 1. Khởi tạo:

*δ1(i) = πibi(o1­) 1 ≤ i ≤ N*

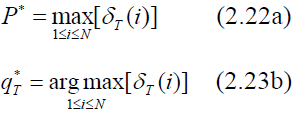
**(2.20**)

*ψ1(i) = 0*

Bước 2. Vòng lặp:



Bước 3. Kết thúc:



Bước 4. Quay ngược lại tìm đường đi (dãy trạng thái):



B.3. Thuật toán thứ ba: Ước lượng tham số của mô hình

Cần phải điều chỉnh các tham số của mô hình như thế nào để P(O|λ) đạt cực đại. Đây là bài toán tối ưu tham số mô hình để mô tả tốt nhất dãy quan sát cho trước. Dãy quan sát dùng để hiệu chỉnh các tham số của mô hình được gọi là dãy huấn luyện. Bài toán huấn luyện là bài toán cốt lõi nhất trong 3 bài toán vì nó cho phép ta tối ưu các tham số mô hình với dữ liệu huấn luyện, tức là tạo ra mô hình tốt nhất cho hiện tượng thực tế.

Không có cách nào giải quyết tập tham số mô hình để làm cực đại xác suất của dãy quan sát một cách chặt chẽ. Tuy nhiên, ta có thể hiệu chỉnh các tham số của mô hình λ = (A, B, π) để xác suất P(O|λ) đạt cực đại địa phương bằng cách dùng phương pháp ước lượng Baum-Welch, còn gọi là phương pháp cực đại kỳ vọng (expectation maximization – EM).

Để mô tả thủ tục ước lượng lại tham số mô hình, đầu tiên ta định nghĩa *ξt(i,j)* là xác suất có trạng thái i tại thời điểm t và trạng thái j tại thời điểm t+1 với mô hình λ và dãy quan sát O đã cho, tức là:

*ξt(i,j) = P(q­t = i, qt+1 = j|O,λ)* (2.25)

Từ định nghĩa các biến tiến và biến lùi trong bài toán 1, ta có thể viết lại *ξt(i,j)* dưới dạng như sau:

 (2.26)





Định nghĩa *γt(i)* là xác suất trạng thái i ở thời điểm t với dãy quan sát O và mô hình λ, vì vậy ta có mối liên hệ giữa *γt(i)* với *ξt(i,j)* như sau:

 (2.27)

Nếu lấy tổng *γt(i)* theo thời gian t, ta nhận được đại lượng chính là kỳ vọng số lần hệ thống ở trạng thái i, hay tương đương với kỳ vọng số chuyển đổi trạng thái được thực hiện từ trạng thái i (nếu bỏ qua t = T trong phép lấy tổng). Tương tự, lấy tổng *ξt(i,j)*theo t (từ t = 1 đến t = T-1) là kỳ vọng số lần chuyển đổi từ trạng thái i sang trạng thái j, tức là :

 kỳ vọng số chuyển trạng thái từ trạng thái i trong O (2.28)

kỳ vọng số chuyển đổi từ trạng thái i sang trạng thái j trong O (2.29)

Từ các công thức trên ta có thể đưa ra một phương pháp đánh giá lại tham số của một mô hình Markov ẩn như sau:

*Ước lượng tham số π :*

* =* kỳ vọng số lần ở trạng thái i tại thời điểm t=1

=  (2.30)

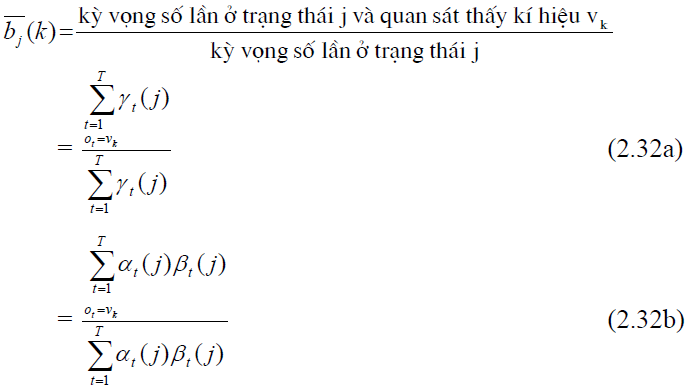


*Ước lượng tham số A :*

** (2.31a)

** (2.31b)

*Ước lượng tham số B :*

**

C. Cấu trúc các gói trong thư viện LIA-RAL

C.1. Cấu trúc file NormFeat.cfg

*\* NormFeat config File*

*mode norm*

*bigEndian false*

*loadFeatureFileFormat SPRO4*

*saveFeatureFileFormat SPRO4*

*loadFeatureFileExtension .prm*

*saveFeatureFileExtension .norm.prm*

*featureServerBufferSize ALL\_FEATURES*

*featureFilesPath ./prm/*

*sampleRate 100*

*saveFeatureFileSPro3DataKind FBCEPSTRA*

*labelSelectedFrames speech*

*segmentalMode false*

*writeAllFeatures true*

*labelFilesPath ./lbl/*

*frameLength 0.01*

*featureServerMode FEATURE\_WRITABLE*

*featureServerMemAlloc 50000000*

C.2. Cấu trúc file NormFeat\_energy.cfg

*\* NormFeat config File*

*mode norm*

*bigEndian false*

*loadFeatureFileFormat SPRO4*

*saveFeatureFileFormat SPRO4*

*loadFeatureFileExtension .prm*

*saveFeatureFileExtension .enr.prm*

*featureServerBufferSize ALL\_FEATURES*

*featureFilesPath ./prm/*

*sampleRate 100*

*saveFeatureFileSPro3DataKind FBCEPSTRA*

*labelSelectedFrames all*

*addDefaultLabel true*

*defaultLabel all*

*segmentalMode false*

*writeAllFeatures true*

*labelFilesPath ./lbl/*

*frameLength 0.01*

*vectSize 60*

*featureServerMode FEATURE\_WRITABLE*

*featureServerMemAlloc 50000000*

### C.3. Cấu trúc file EnergyDetector.cfg

*\*\*\* EnergyDetector Config File*

*\*\*\**

*loadFeatureFileExtension .enr.prm*

*minLLK -200*

*maxLLK 1000*

*bigEndian false*

*loadFeatureFileFormat SPRO4*

*saveFeatureFileFormat SPRO4*

*saveFeatureFileSPro3DataKind FBCEPSTRA*

*featureServerBufferSize ALL\_FEATURES*

*featureServerMemAlloc 50000000*

*featureFilesPath ./prm/*

*mixtureFilesPath ./*

*lstPath ./lst/*

*labelOutputFrames speech*

*labelSelectedFrames all*

*addDefaultLabel true*

*defaultLabel all*

*saveLabelFileExtension .lbl*

*labelFilesPath ./lbl/*

*frameLength 0.01*

*segmentalMode file*

*nbTrainIt 8*

*varianceFlooring 0.0001*

*varianceCeiling 1.5*

*alpha 0.0*

*mixtureDistribCount 3*

*featureServerMask 19*

*vectSize 1*

*baggedFrameProbabilityInit 0.001*

*thresholdMode weight*

*Cấu trúc file TrainWorldInit.cfg*

*distribType GD*

*mixtureDistribCount 20*

*loadMixtureFileExtension .gmm*

*saveMixtureFileExtension .gmm*

*loadFeatureFileExtension .norm.prm*

*maxLLK 200*

*minLLK -200*

*bigEndian false*

*saveMixtureFileFormat RAW*

*loadMixtureFileFormat RAW*

*loadFeatureFileFormat SPRO4*

*featureServerBufferSize ALL\_FEATURES*

*featureServerMemAlloc 100000000*

*featureFilesPath ./prm/*

*mixtureFilesPath ./gmm/*

*labelFilesPath ./lbl/*

*frameLength 0.01*

*lstPath ./lst/*

*labelSelectedFrames speech*

*baggedFrameProbability 0.2*

*baggedFrameProbabilityInit 0.5*

*initVarianceFlooring 0.5*

*initVarianceCeiling 1*

*finalVarianceFlooring 0.5*

*finalVarianceCeiling 5.0*

*normalizeModel true*

*nbTrainIt 10*

*use01 true*

*verbose true*

*featureServerMask 0-18,20-50*

*baggedMinimalLength 1*

*baggedMaximalLength 1*

C.4. Cấu trúc file TrainWorldInit.cfg

*distribType GD*

*mixtureDistribCount 20*

*loadMixtureFileExtension .gmm*

*saveMixtureFileExtension .gmm*

*loadFeatureFileExtension .norm.prm*

*maxLLK 200*

*minLLK -200*

*bigEndian false*

*saveMixtureFileFormat RAW*

*loadMixtureFileFormat RAW*

*loadFeatureFileFormat SPRO4*

*featureServerBufferSize ALL\_FEATURES*

*featureServerMemAlloc 100000000*

*featureFilesPath ./prm/*

*mixtureFilesPath ./gmm/*

*labelFilesPath ./lbl/*

*frameLength 0.01*

*lstPath ./lst/*

*labelSelectedFrames speech*

*baggedFrameProbability 0.2*

*baggedFrameProbabilityInit 0.5*

*initVarianceFlooring 0.5*

*initVarianceCeiling 1*

*finalVarianceFlooring 0.5*

*finalVarianceCeiling 5.0*

*normalizeModel true*

*nbTrainIt 10*

*use01 true*

*verbose true*

*featureServerMask 0-18,20-50*

*baggedMinimalLength 1*

*baggedMaximalLength 1*

C.5. Cấu trúc file TrainWorldFinal.cfg

*distribType GD*

*mixtureDistribCount 20*

*loadMixtureFileExtension .gmm*

*saveMixtureFileExtension .gmm*

*loadFeatureFileExtension .norm.prm*

*maxLLK 200*

*minLLK -200*

*bigEndian false*

*saveMixtureFileFormat RAW*

*loadMixtureFileFormat RAW*

*loadFeatureFileFormat SPRO4*

*featureServerBufferSize ALL\_FEATURES*

*featureServerMemAlloc 100000000*

*featureFilesPath ./prm/*

*mixtureFilesPath ./gmm/*

*labelFilesPath ./lbl/*

*frameLength 0.01*

*lstPath ./lst/*

*labelSelectedFrames speech*

*baggedFrameProbability 1*

*initVarianceFlooring 0.5*

*initVarianceCeiling 5.0*

*finalVarianceFlooring 0.5*

*finalVarianceCeiling 5.0*

*featureServerMask 0-18,20-50*

*normalizeModel true*

*use01 true*

*nbTrainIt 2*

*baggedMinimalLength 1*

*baggedMaximalLength 1*

C.6. Cấu trúc gói TrainTarget

*\*\*\* TrainTarget Configuration File*

*\*\*\**

*maxLLK 200*

*minLLK -200*

*bigEndian false*

*saveMixtureFileFormat RAW*

*loadMixtureFileFormat RAW*

*loadFeatureFileFormat SPRO4*

*featureServerBufferSize ALL\_FEATURES*

*loadMixtureFileExtension .gmm*

*saveMixtureFileExtension .gmm*

*loadFeatureFileExtension .norm.prm*

*featureFilesPath ./prm/*

*labelFilesPath ./lbl/*

*mixtureFilesPath ./gmm/*

*nbTrainIt 1*

*labelSelectedFrames speech*

*normalizeModel true*

*normalizeModelMeanOnly true*

*normalizeModelNbIt 5*

*MAPAlgo MAPOccDep*

*meanAdapt true*

*MAPRegFactorMean 14*

*frameLength 0.01*

*featureServerMemAlloc 100000000*

*featureServerMask 0-18,20-50*

C.7. Cấu trúc gói ComputeTest

*\*\*\* ComputeTest Config File*

*\*\*\**

*distribType GD*

*loadMixtureFileExtension*

*saveMixtureFileExtension .gmm*

*loadFeatureFileExtension .prm*

*mixtureDistribCount 1024*

*maxLLK 200*

*minLLK -200*

*bigEndian false*

*saveMixtureFileFormat RAW*

*loadMixtureFileFormat RAW*

*loadFeatureFileFormat SPRO3*

*featureServerBufferSize ALL\_FEATURES*

*featureFilesPath ./*

*mixtureFilesPath ./*

*labelSelectedFrames male*

*labelFilesPath ./*

*frameLength 0.01*

*segmentalMode segmentLLR*

*topDistribsCount 10*

*computeLLKWithTopDistribs COMPLETE*

*ndxFileName ndx*

*worldModelName wld*

*outputFile test1.res*

*gender M*

*debug true*

*verbose true*

*featureServerMask 0-15,17-32*

*vectSize* 32

D. Code chương trình nhận dạng người nói tiếng Việt

D.1. Code chương trình tổng hợp xác suất của ALIZE và Sphinx