BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

NGUYỄN HOÀNG HẢI

HỆ THỐNG NHẬN DẠNG CHỮ VIẾT TAY TIẾNG VIỆT IN HOA TRỰC TUYẾN

LUẬN VĂN THẠC SĨ KHOA HỌC

Kỹ thuật điện tử viễn thông

Hà Nội - 2012

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

NGUYỄN HOÀNG HẢI

HỆ THỐNG NHẬN DẠNG CHỮ VIẾT TAY TIẾNG VIỆT IN HOA TRỰC TUYẾN

Chuyên ngành: Kỹ thuật điện tử viễn thông

LUẬN VĂN THẠC SĨ KHOA HỌC

Kỹ thuật điện tử viễn thông

NGƯỜI HƯỚNG DẪN KHOA HỌC:
TS. PHẠM NGỌC NAM

Hà Nội - 2012

LỜI CAM ĐOAN

Học viên xin cam đoan đây là công trình nghiên cứu độc lập của mình. Luận văn được hoàn thành bởi học viên dưới sự hướng dẫn của TS.Phạm Ngọc Nam.

Các tài liệu tham khảo, công thức, sử dụng trong luận văn đều được ghi nguồn gốc cụ thể, rõ ràng.

Hà Nội, ngày tháng 03 năm 2012

Học viên cao học

Nguyễn Hoàng Hải

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

HMM	Hidden Markov Model
Stroke	Nét vẽ ký tự
NNA	Neural network approach
ChainCode	Chuỗi từ mã mô tả quỹ đạo viết
OHWR	Online Hand writing Regconition
Input Device	Thiết bị đầu vào
Feather extractor	Trích lọc các thuộc tính
Preprocessor	Khỗi tiền xử lý
Postprocessor	Hậu xử lý
Character	Ký tự nhận dạng.
OCR	Optical character recognition
MEMM	Maximum Entropy Markov Model
IBM	International Business Machines Corporation
FORTRAN	Formula Translator/Translation
LCD	liquid crystal display
ANN	Artificial neural network

TDNN	Time delay neural network
picturebox	Một đối tượng sử dụng trên bộ lập trình Visual studio
Tablet	Thiết bị cầm tay cảm ứng
Raw data	Lọc dữ liệu
Point distance	Đánh giá khoảng cách giữa các điểm
Segmentation	Phân đoạn dữ liệu

DANH MỤC CÁC HÌNH VẼ

LỜI MỞ ĐẦU

Bài toán nhận dạng chữ viết được ra đời từ thập niên 50 của thế kỷ trước, cho đến nay vẫn nhận được nhiều sự quan tâm của các nhà nghiên cứu khoa học trên thế giới. Đặc biệt trong những năm gần đây cùng với sự phát triển mạnh mẽ của các thiết bị cảm ứng, đã đặt ra một thách thức đối với khoa học nhận dạng. Trên thế giới, hầu hết các thiết bị điện tử cảm ứng mới ra đời đều đi kèm với nó các phần mềm nhận dạng chữ viết tay, hàng loạt các nhà sản xuất lớn như Microsoft, Apple,.. đều tích hợp module nhận dạng chữ viết vào các hệ thống của họ để thu hút người dùng, điều đó cho thấy sự cần thiết của khoa học nhận dạng chữ viết đối với đời sống con người hiện nay. Ở Việt Nam, Các nhà khoa học cũng không đứng ngoài xu hướng phát triển đó, đã có một số các phần mềm nhận dạng chữ viết ra đời như phần mềm nhận dạng chữ viết tiếng việt ABBYY, phần mềm nhận dạng chữ in OCR (Optical character recognition)….Các nghiên cứu này nhằm tối ưu các kỹ thuật nhận dạng đối với chữ viết tiếng việt. Tuy nhiên lĩnh vực nhận dạng chữ viết tay, hiện nay được chia thành hai hướng : nhận dạng chữ viết online và nhận dạng chữ viết offline. Hiện nay các nhà khoa học Việt Nam mới chủ yếu tập trung nghiên cứu nhận dạng chữ viết offline- nhận dạng trên các hình vẽ hay các bản viết tay, còn các nghiên cứu nhận dạng chữ viết online- nhận dạng theo hướng của người viết được ứng dụng nhiều đối với các thiết bị cảm ứng- hiện tại còn khá ít nghiên cứu. Trong luận văn này tác giả tập trung nghiên cứu theo hướng nhận dạng chữ viết online, mục tiêu của đồ án này là đưa ra một cách tiếp cận trong việc nhận dạng chữ viết online dựa trên mô hình Markov. Cấu trúc luận án được chia làm 5 phần: Chương I giới thiệu về lịch sử và sự cần thiết của khoa học nhận dạng, Chương II giới thiệu tổng quan về hệ thống nhận dạng chữ viết, Chương III lý thuyết giải thuật Markov, Chương IV ứng dụng giải thuật Markov trong việc nhận dạng chữ viết online, Phần cuối là kết luận và hướng phát triển tiếp theo của luận văn.

Qua đây tôi xin gửi lời cảm ơn chân thành đến TS.Phạm Ngọc Nam đã tận tình hướng dẫn tôi trong suốt thời gian làm luận văn. Tôi cũng xin gửi lời cảm ơn đến các bạn bè, gia đình và đồng nghiệp đã giúp đỡ tôi hoàn thiện luận văn này.

CHƯƠNG 1 : TỔNG QUAN VỀ NHẬN DẠNG CHỮ VIẾT

Chương này sẽ giới thiệu về lich sử của khoa học nhận dạng và sự cần thiết của hệ thống nhận dạng nói chung và hệ thống nhận dạng chữ viết nói riêng.Cũng trong chương này chúng ta phân loại các hệ thống nhận dạng chữ viết và có cái nhìn tổng quan về một hệ thống nhận dạng chữ viết.

1.1 Lịch sử và sự cần thiết của ngành khoa học nhận dạng.

Từ giữa những năm 50 của thế kỷ XX, nhận dạng là một lĩnh vực có sức hấp dẫn rất lớn cho việc nghiên cứu và phát triển, thu hút nhiều nhà khoa học tham gia nghiên cứu. Những sản phẩm nhận dạng chữ viết mang tính thương mại bắt đầu xuất hiện vào những năm 60. Một vài phương pháp nhận dạng đơn giản đã được đưa vào áp dụng. Có thể kể một vài sản phẩm tiêu biểu là: IBM 1418, IBM 1428, IBM 1285, IBM 1287 của IBM, Facom 6399 của Fuitsu và H-852 của Hitachi. Vào những năm 70, Tiến bộ về công nghệ cho phép chế tạo các máy quét laser giá rẻ có chất lượng cao, một số phần mềm khác cũng được giới thiệu, được viết chủ yếu trên nền FORTRAN, sản phẩm nâng cấp H8959 của Hitachi đã đạt tới độ chính xác 100% nếu sử dụng mẫu chữ một người và thử nghiệm lại bằng chính nét chữ của người đó. Đặc biệt là trong những năm gần đây, cùng với sự bùng nổ các thiết bị cảm ứng cầm tay, nhận dạng không chỉ còn là lĩnh vực nghiên cứu lý thuyết nữa mà đã được ứng dụng rộng rãi trong thực tế cuộc sống. Các bài toán nhận dạng được nghiên cứu nhiều nhất hiện nay bao gồm nhận dạng các mẫu hình học (vân tay, mặt người, hình khối,…), nhận dạng tiếng nói và nhận dạng chữ viết. Chúng được áp dụng vào nhiều lĩnh vực như y học, dự báo thời tiết, dự báo cháy rừng, điều khiển robot, ... Trong đó bài toán nhận dạng chữ viết là bài toán có sức thu hút hơn cả bởi sự bùng nổ các thiết bị cầm tay cảm ứng hiện nay.

Bạn hãy tưởng tượng trong tương lai mà một máy tính không cần bàn phím. Thay vào đó tất cả các lệnh và dữ liệu được cho vào máy tính một cách rất tự nhiên: bằng văn bản. Khi đó tất cả các công việc tẻ nhạt của con người chẳng hạn như đọc kiểm tra trong hệ thống ngân hàng, phân loại thư trong hệ thống bưu chính... được thay thế bằng máy. điều này chỉ có thể thực hiện được khi máy tính hiểu được các ký tự bạn viết trên văn bản đó, máy tính nhận dạng được các ký tự viết tay của bạn. Đó là một thực tế để thấy được tiềm năng to lớn của lĩnh vực nhận dạng chữ viết.

Trong bối cảnh các thiết bị cảm ứng như smart phone, Ipad… ngày càng trở nên phổ biến, người ta có thể viết trực tiếp trên một màn hình tinh thể lỏng (LCD) hiển thị với một bút điện tử hay đơn đơn giản là ngón tay. Màn hình là một ma trận nhạy cảm ghi lại sự chuyển động của đầu bút trên bề mặt. Quỹ đạo của các cây bút xuất hiện gần như ngay lập tức trên màn hình. Nhận dạng chữ viết dựa trên quỹ đạo bút khi đó đóng vai trò quan trọng cho phép máy tính hiểu người viết muốn thực hiện thao tác gì, tập lệnh gì .... Việc nhận dạng chữ viết trong trường hợp này có ưu thế hơn so với hình thức nhận dạng tiếng nói ở tính bảo mật.

Trong hệ thống ngân hàng, các biểu mẫu thuế tự động và máy đọc tự động là những ứng dụng rất hấp dẫn nhận dạng chữ viết. Loại ứng dụng này có đặc điểm số lượng từ vựng sử dụng ít. Các từ vựng khoảng 35 từ và số, như đơn vị tiền tệ….Nếu sử dụng loại máy này sẽ tiết kiệm chi phí về thời gian và nhân lực. Tuy nhiên nó đòi hỏi một sự công nhận tỷ lệ rất cao mà đây vẫn là một trở ngại trong hệ thống nhận dạng chữ viết tay.

Trong các ứng dụng bưu chính, với đặc điểm sử dụng vốn từ vựng lớn, có chứa tất cả đường phố, thành phố, quận huyện, và quốc gia tên. Hiện nay, có tồn tại các máy tự động có thể sắp xếp thư dựa đọc mã zip. Tuy nhiên trong nhiều trường hợp, một hoặc nhiều chữ số trong các mã zip được nhận dạng sai có thể gây ra phân loại sai. Để cải thiện hiệu suất của máy phân loại, công nhận tên đường phố và thành phố nên được kết hợp với nhận dạng mã vùng.

Một ứng dụng quan trọng của nhận dạng chữ viết là sự sao chép của văn bản dạng chữ viết tay cho phép máy tính có thể đọc được văn bản. Rõ ràng, văn bản

được viết bởi một cây bút và tờ giấy nhanh hơn và dễ dàng hơn nhiều nếu gõ từ bàn phím, đặc biệt là đối với ngôn ngữ như Trung Quốc, Lào, Thái Lan…. Các ứng dụng sao chép văn bản là rất lớn. Ví dụ trong lưu trữ hệ thống để lưu trữ các tài liệu lưu trữ hàng trăm năm tuổi, dữ liệu sẽ được quét thành các file ảnh và sau đó được lưu trữ trong các thiết bị lưu trữ như đĩa cứng và đĩa CD ROM. Tuy nhiên ngay cả đối với các tập tin hình ảnh định dạng tốt nhất, kích thước vẫn còn rất lớn so với kích thước của một tập tin văn bản. Với hệ thống sao chép văn bản tất cả các tài liệu đầu tiên được quét và sau đó chuyển đổi và được lưu trữ trong các hình thức của các tập tin văn bản, yêu cầu không gian ít hơn. Một hệ thống như vậy đem lại lợi ích kinh tế rất lớn.

Rõ ràng, tất cả các ứng dụng trên phụ thuộc mạnh mẽ vào tính chính xác của hệ thống nhận dạng. Câu hỏi đặt ra là chúng ta đã xây dựng được một hệ thống đáp ứng được các nhu cầu thực tế đó chưa? Câu trả lời chúng ta đã có những hệ thống nhận dạng tuy nhiên tính chính xác tuyệt đối vẫn còn cần thời gian nghiên cứu thêm . Chỉ có một vài năm trước đây, nhận dạng chữ viết chữ thảo dường như ngoài tầm với, ngày nay giấc mơ đã trở thành hiện thực thực. Nhiều hệ thống nhận dạng ra đời với độ chính xác cao đáp ứng được một số các nhu cầu cơ bản không đòi hỏi tính chính xác tuyệt đối.

1.2 Phân loại nhận dạng chữ viết

Nhận dạng chữ viết bao gồm hai kiểu chính là nhận dạng chữ in và nhận dạng chữ viết tay. Cho đến nay bài toán nhận dạng chữ in đã được giải quyết khá trọn vẹn với sự ra đời của nhiều hệ thống nhận dạng đạt tới độ chính xác gần như tuyệt đối. Tiêu biểu có hệ nhận dạng chữ in dựa trên mô hình mạng nơron bốn lớp của J. Wang và J.S.N. Jean. Ở Việt Nam hiện đã có sản phẩm VNDOCR của Viện Công nghệ thông tin nhận dạng chữ in tiếng Việt.

Nhận dạng chữ viết tay được chia thành hai lớp bài toán lớn là nhận dạng chữ viết tay online và nhận dạng chữ viết tay offline. Trong nhận dạng chữ viết tay offline, dữ liệu đầu vào được cho dưới dạng các ảnh được quét từ các giấy tờ, văn bản. Các mẫu bit của hình ảnh bằng văn bản như chuyển đổi bởi một máy quét quang học.

Ngược lại nhận dạng chữ viết tay trực tuyến là nhận dạng dựa trên quỹ đạo người viết hệ thống lấy ra được số nét vẽ, hướng viết ….

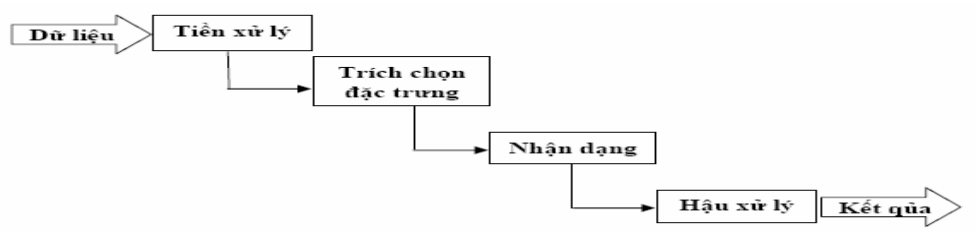
Phong cách viết trong các hệ thống nhận dạng.

Trong hệ thống nhận dạng một phong cách viết, các hệ thống được xây dựng để nhận ra các văn bản của phong cách viết cụ thể. Tỷ lệ nhận dạng của hệ thống là vì thế cao cho đối với người viết đó và sẽ giảm xuống đáng kể cho những phong cách viết khác. Trong hệ thống đa phong cách viết, các hệ thống được xây dựng để nhận dạng với chữ viết tay của hàng trăm phong cách viết khác nhau. Do đó tỷ lệ nhận dạng đối với từng phong cách viết thấp hơn so với hệ thống đơn phong cách. Tùy thuộc vào ứng dụng, hệ thống nhận dạng có thể được xây dựng dựa trên một trong hai cách trên. Đối với một thiết bị cầm tay, Sẽ tối ưu nếu chúng ta xây dựng hệ thống đơn phong cách, trong khi đối với văn bản sao chép các ứng dụng hệ thống đa phong cách thực sự là một lựa chọn tốt hơn..

Nhận dạng chữ viết tay in thường và chữ viết tay theo phong cách bản thảo

Các phong cách viết có thể được chia thành hai loại chữ viết tay in và chữ viết tay chữ thảo. Tuy nhiên, trong thực tế, ranh giới giữa hai phong cách của văn bản là không rõ ràng. Nhận dạng chữ viết tay chữ thảo rất nhiều khó khăn hơn so với dạng chữ viết in. Chữ viết in thường được người viết xử lý cẩn thận do vậy các đặc tính của chữ viết không bị biến dạng. Trong khi đó, với chữ viết tay chữ thảo, người viết có thể viết tự do ở các tốc độ khác nhau. Đối với loại chữ viết tay, Ngay cả con người cũng khó khăn trong việc nhận dạng. Do vậy khi áp dụng trên hệ thống sẽ có những sai xót nhất định.

1.3 Mô hình tổng quan hệ thống nhận dạng chữ viết



Hình 1-1 Mô hình chung hệ thống nhận dạng chữ viết

Trong đó:

- Tiền xử lý: là quá trình chuẩn hóa dữ liệu vào, gồm các công việc như xử lý nhiễu, chuẩn hóa kích thước dữ liệu, …
- Trích chọn đặc trưng: là quá trình tìm ra các thông tin hữu ích và đặc trưng nhất cho mẫu đầu vào để sử dụng cho quá trình nhận dạng.
- Nhận dạng: là quá trình sử dụng một mô hình nhận dạng cụ thể với một thuật toán cụ thể để trả lời mẫu đầu vào là ký tự nào.
- Hậu xử lý: là quá trình xử lý kết quả cho phù hợp với từng ứng dụng cụ thể.

1.4 Tóm tắt thông tin

Chương này giới thiệu về lịch sử phát triển của lĩnh vực nhận dạng nói chung và nhấn mạnh sự cần thiết của nhận dạng chữ viết nói riêng. Giới thiệu mô hình tổng quan của một hệ thống nhận dạng dưới dạng sơ đồ logic, chức năng từng khối. Trong chương sau chúng ta sẽ tìm hiểu sâu hơn về các vấn đề khi nhận dạng chữ viết online.

CHƯƠNG 2: NHẬN DẠNG CHỮ VIẾT ONLINE

Như đã mô tả tổng quan ở chương I, Một hệ thống nhận dạng bao gồm các khối logic tiền xử lý, trích chọn đặc trưng, nhận dạng và hậu xử lý. Tiền xử lý đóng vai trò quan trọng trong việc nâng cao tính chính xác của hệ thống, trích chọn đặc trưng lấy ra các thông tin quan trọng trên quỹ đạo bút của người viết như số nét vẽ, các điểm vẽ được lấy ra và tạo thành các chuỗi mã biểu diễn dữ liệu, khối nhận dạng thực hiện nhiệm vụ chính là của hệ thống bằng cách sử dụng một hoặc nhiều phương pháp có sẵn như Artificial neural network (ANN), Hidden Markov Model (HMMs)... hậu xử lý đưa kết quả việc nhận dạng cho người sử dụng. Các chi tiết của các giai đoạn khác nhau trong hệ thống nhận dạng chữ viết online được trình bày trong chương này. Trước hết, các đặc điểm nhận dạng chữ viết tay trực tuyến được thảo luận.

2.1. Các đặc điểm nhận dạng chữ viết online

2.1.1Thu thập dữ liệu đầu vào

Nhận dạng chữ viết online có đặc điểm thú vị là ta có thể nhận biết được quỹ đạo của nét bút người viết, tốc độ viết, số lượng các nét viết, thứ tự các nét. Một nét vẽ được tính từ khi bút đặt xuống để viết đến khi nhấc lên. Khi viết, việc nhận ghi dữ liệu có thể được thu thập theo hai cách khác nhau. Trong cách đầu tiên, quỹ đạo bút được chuyển đổi thành hình ảnh điểm ảnh và xử lý với nhận diện ký tự in (OCR) nhận diện như trong nhận dạng off-line. Trong cách thứ hai, bút quỹ đạo được sử dụng như trình tự của các điểm toạ độ (x, y). Việc lấy dữ liệu theo cách 2 có đặc điểm quan trọng sau:

- Biết được Trình tự viết và có thể được sử dụng bởi quá trình nhận dạng
- Thông tin như tốc độ cũng có thể được xem xét như một yếu tố đầu vào
- Thông tin số lần nhấc bút đươc sử dụng hiệu quả để nhận biết số nét vẽ hoặc phân đoạn ký tự viết.

Hình 2.1 cho thấy sự khác biệt của nhận dạng online và offline. Đối với nhận dạng online hệ thống sẽ biết được trình tự của người viết và hiểu rằng chữ H ở dưới có hai cách viết, trong khi đối với nhận dạng offline chỉ nhận biết được một cách viết.

Hình 2- 1 Hai cách viết khác nhau của chữ in H

2.1.2Các thuộc tính chữ viết tay

Trong phần này, các thuộc tính dạng chữ viết tay của ngôn ngữ tiếng Việt sẽ được thảo luận.Tiếng Việt bao gồm 33 chữ cái và các dấu chấm câu. Bảng chữ cái tiếng Việt mỗi chữ cái có hai hình thức viết: chữ viết thường và chữ viết hoa. Một từ được cấu thành bởi các chữ cái bao gồm các trình tự của các chữ cái theo thứ tự không gian từ trái sang phải. Trong tiếng việt mỗi một chữ cái có đặc điểm khác nhau, về số nét vẽ, về trình tự viết, về quỹ đạo nét bút khi viết việc quan sát các đặc điểm này sẽ giúp tăng tính chính xác của hệ thống và tăng hiệu năng của hệ thống ví dụ chữ Â được cấu thành bởi nét vẽ, viết từ trái sang phải…

2.1.3Một số vấn đề khi nhận dạng

Còn nhiều vấn đề khi nhận dạng chữ viết trên bao gồm nhận dạng ký tự, nhận dạng từ, nhận dạng các chữ số và các ký hiệu khác... Những khó khăn của của việc nhận dạng do một số các yêu tố:

–Hạn chế về số lượng người viết.

–Hạn chế về cách viết, một người viết có thể viết cùng một ký tự theo các cách khác nhau.

–Khó khăn về ngôn ngữ: giới hạn về số ký tự nhận dạng, kích thước của từ, nét vẽ.

– tồn tại nhiều cặp ký tự có hình dạng tương tự, gây khó khăn để nhận biết. ví dụ các cặp: UV, CL, Oo, I-1, l-1, Z-2, S-5, G-6, Cc, Kk... Đối với hệ thống nhận dạng dựa trên đặc tính ngữ cảnh sử dụng mới có thể phân biệt được các ký tự này.

2.3 Tiền xử lý

Trong nhận dạng chữ viết online, tiền xử lý đóng một vai trò quan trọng quyết định hiệu năng và tính chính xác của hệ thống nhận dạng. Tiền xử lý dữ liệu giúp lấy ra các thông tin quan trọng và loại bỏ các dữ liệu dư thừa đối với hệ thống nhận dạng. Dữ liệu thu được trên giấy, thiết bị cảm ứng, hay đối tượng picturebox của visualstudio luôn luôn có các nét dư thừa do hiệu ứng lượng tử hóa số hóa, tay chuyển động không chính xác khi viết. Thông thường, tiền xử lý bao gồm lọc các dữ liệu xấu, làm mịn nét vẽ, lấy mẫu và phân đoạn. Mỗi của những công việc này sẽ được thảo luận ngắn gọn.

2.3.1 Lọc dữ liệu

Mục đích của lọc dữ liệu là để loại bỏ các dữ liệu xấu, các dữ liệu dư thừa không cần thiết cho quá trình nhận dạng, mà còn giảm hiệu năng và tính chính xác của hệ thống. Việc lọc bỏ trong luận văn được tác giả sử dụng bằng cách lấy ra toàn bộ các điểm của ký tự quan sát đầu vào, sau đó lấy ra một số điểm trong tập các điểm thu được sao cho khoảng cách giữa hai điểm nhỏ hơn một khoảng cách được chọn phù hợp với kích thước của khổ giấy, khung của thiết bị cầm tay …

Smoothing: làm mịn làm giảm sự hiện diện lỗi số hóa. Một kỹ thuật làm mịn thông thường là trung bình một điểm với các điểm lân cận của nó.

De-hooking: de-móc loại bỏ các móc có thể xuất hiện ngay thời điểm ban đầu nhưng thường xuyên tại điểm kết thúc của nét vẽ. Các móc phụ thuộc vào vị chí đặt bút (chuột máy tính) và tốc độ viết, người viết, bề mặt thiết bị cầm tay nếu có.

2.3.2 Lấy mẫu

Mục tiêu của lấy mẫu là để loại bỏ chênh lệch bằng văn bản. Những chênh lệch này có thể là nghiêng của văn bản, kích thước của văn bản và như vậy. Kích thước lấy mẫu: Cách đơn giản nhất của việc lấy mẫu thường được dùng là re-scaling. Nếu một ký tự được viết chính xác như nhau hai lần với ngoại lệ của một trong những ký tự xuất hiện nhiều cao hơn hoặc rộng hơn, cả hai ký tự cần được phân loại giống nhau và với một mức độ cao. Sự khác biệt trong kích thước của các ký tự bằng văn bản nên được loại bỏ. Nó không được sử dụng cho quá trình phân loại. Để bảo toàn

hình dạng của một ký tự sau khi thay đổi, cả hai kích thước (ngang và dọc) phải được mở rộng với tỷ lệ như nhau. Một thuật toán được sử dụng thường được dựa trên khung giới hạn của mỗi ký tự đã được phân đoạn. Ranh giới được xác định bởi tọa độ (x,y) lớn nhất và bé nhất của ký tự. Mở rộng được thực hiện bằng cách áp dụng việc chuyển đổi sau mỗi cặp x, y tọa độ: x '= cx, y' = cy c là các tham số mở rộng phụ thuộc vào chiều cao và chiều rộng của hình bao ký tự.

Hình 2-2 Đường bao chữ cái a

Độ nghiêng: mức độ nghiêng khi viết ký tự phụ thuộc vào người viết. Một thuật toán giải quyết độ nghiêng được áp dụng để xử lý việc này.

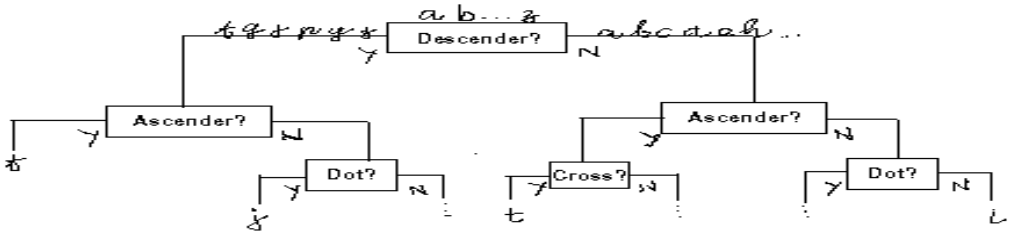
2.4 Chiết xuất các thuộc tính

Khối chiết xuất các thuộc tính đóng vai trò quan trọng nhất trong hệ thống nhận dạng. Nó quyết định hiệu năng của toàn bộ hệ thống, tính chính xác của hệ thống. Ký tự sau khi được viết lên giấy, tablet hay picturebox sẽ được khối này tạo ra thành các từ mã cùng với dữ liệu về số nét vẽ sẽ là đầu vào của việc xử lý và nhận dạng bước tiếp theo.

2.4.1 Cây nhị phân

Như tên gọi, các tính năng có thể được nhị phân, ví dụ, descender hoặc no descender không có dấu chấm hoặc không có dấu chấm, chéo hoặc không chéo. Tên gán cho ký tự không rõ thường được xác định bởi một cây quyết định, mà có thể được tạo ra như thể hiện trong hình 2.3. Sự hiện diện của descender làm giảm sự lựa chọn f, g, k, p, q, y, z. Sau đó, nếu một dấu chấm là hiện tại, sự lựa chọn duy nhất là j. Trong thực tế, cây quyết định nhị phân thường được sử dụng thêm các tính năng khác. Cây nhị phân quyết định chỉ sử dụng tính năng đơn giản để giảm sự tập hợp

của các nhân vật ứng cử viên cho một tập hợp nhỏ để phân tích tiếp theo bởi các tính năng phức tạp.



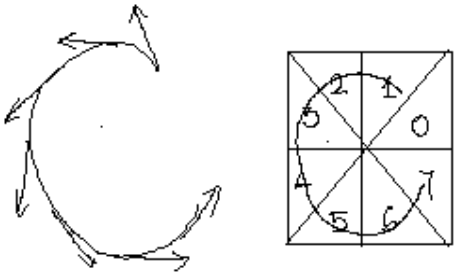
Hình 2- 3 Cây quyết định

2.4.2 Chuỗi mã

Một ký tự có thể được mô dưới dạng một tập hợp các điểm và hướng của ký tự (trái, phải, lên, xuống). Trình tự như vậy được gọi là chuỗi mã. Ví dụ Chữ C có thể được đại diện bởi một chuỗi các mã chuỗi: lên-trái-xuống-xuống-xuống-phải-lên.

2.4.3 Trình tự thời gian của các zone

Một ký tự có thể được đại diện bởi một chuỗi được mã hóa. Các zone được quy định bằng cách chia hình chữ nhật bao quanh các ký tự bằng văn bản. Các ký tự được bao bởi một hình chữ nhật và trình tự của zone được xác định bởi đầu bút hoặc con chuột máy tình. Trình tự, hoặc một chuỗi các tính năng tương ứng, sau đó gán một tên cho một ký tự chưa xác định. Như thể hiện trong hình 2.7 b) chữ C có thể được đại diện bởi một chuỗi các mã vùng 1-2-3-4-5-6-7.

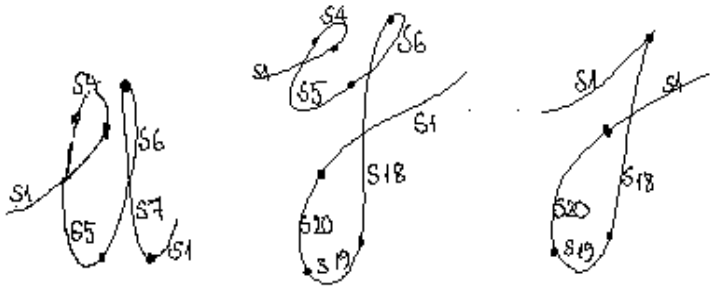


Hình 2- 4 Trình tự viết và phân vùng đề nhận dạng chữ C

2.4.4 Mã hóa nét vẽ

Một ký tự cũng có thể được đại diện bởi một chuỗi các subparts của ký tự thường sử dụng các nét vẽ để mã hóa.. Một chữ cái, g, j được đại diện bởi chuỗi s1-s4 s5-

S6, S7-s1, s1-s4, s5-S6-S18-S19-S20-s1 và s1-S18-S19-S20-s1 tương ứng ví dụ như hình 2.8



Hình 2- 5 Minh họa mã hóa theo nét vẽ

2.4.5 chức năng thuộc tính

Trong tiếng Việt, có một số cặp ký tự có thể gây nhầm lẫn. Ví dụ, phân biệt CO dựa trên điểm đầu và điểm cuối và sự khác biệt VY là một trong những phần mở rộng dòng. Bằng cách sử dụng các thuộc tính chức năng, cặp gây nhầm lẫn có thể phân biệt được.

2.5 Nhận dạng

Nhận dạng liên quan đến việc phân loại từng đối tượng không rõ ràng vào một trong một số hữu hạn các loại. Có hai cách tiếp cận khác nhau để phân loại: phân loại khoảng cách tối thiểu và phân loại khả năng tối đa.

- Phân loại theo khoảng cách tối thiểu. Nếu mô hình từ các cung cùng một lớp học trong một số tính năng không gian, bằng cách sử dụng một số biện pháp về khoảng cách, sau đó các cung này có thể được sử dụng để xác định một số quyết định ranh giới tuyến tính để phân loại một mô hình đến. Một phương pháp là tạo ra vector mẫu lượng của các cung bằng cách tính toán các trung tâm cung. Một mô hình đến sau đó có thể được phân loại bằng cách sử dụng các cung trung tâm là khoảng cách tối thiểu từ các mô hình trên tất cả các cung. Các phương pháp khác có thể chọn một mô hình tốt nhất đại diện cho rằng cụm từ mô hình trong cluster, hoặc có thể chọn một số tập hợp con của những người mẫu để xác định ranh giới quyết định. Tùy thuộc vào

sự phân bố của dữ liệu trong mỗi lớp học, nó có thể là thuận lợi để tạo ra các cụm nhiều cho mỗi lớp. Trong trường hợp này có nhiều mẫu hoặc các trung tâm cụm được lưu trữ để đại diện cho mỗi lớp học, thực hiện phân loại dựa trên các mô hình K gần nhất, hoặc trên K gần nhất trung tâm có thể cung cấp kết quả tốt hơn. Các biện pháp phổ biến khoảng cách là khoảng cách Euclid, khoảng cách Manhattan, và khoảng cách Hamming. Tùy thuộc vào miền ứng dụng, các biện pháp khoảng cách phức tạp hơn nhiều cũng được sử dụng như Ikura Saito khoảng cách, thường được sử dụng trong lĩnh vực nhận dạng giọng nói.

- Phân loại theo xác xuất lớn nhất. Thay vì xác định một số đường biên tuyến tính, một phương pháp tiếp cận khác là đánh giá sự phân bố của dữ liệu, có thể sử dụng phân bố Gaussian, hoặc kết hợp phân bố Gaussian và lý thuyết quyết định Bayes để xây dựng một phương pháp phân loại đến một mô hình dựa trên sự so sánh của một chức năng mật độ xác suất. Sử dụng ví dụ trước các thuộc tính của mỗi lớp (train dữ liệu), chúng ta có thể ước tính xác xuất phân bố có điều kiện của mỗi lớp, P (x | C), trong đó x là một vector tính năng (ví dụ như một chuỗi các chuỗi mã), và C là lớp mẫu sản tạo ra vector tính năng. Sử dụng xác suất của lớp, P (C), và sự xuất hiện của các vector tính năng, P (x) chúng ta có thể tính toán xác suất xuất hiện lớp bằng cách sử dụng các quy tắc quyết định Bayes:

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

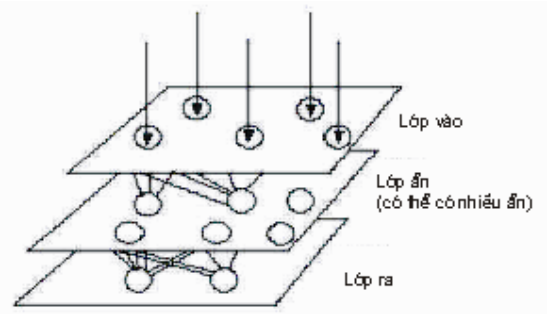
Lớp C có xác xuất P (x | C) lớn nhất được xác định là kết quả của quá trình nhận dạng. Trong tình huống xác suất các lớp bằng nhau (P (C) = 1 / M), ∀ C, trong đó M là số lượng các lớp, các quy tắc quyết định Bayes đơn giản hóa các quy tắc quyết định khả năng tối đa. Chỉ định x lớp C là P (x | C) là tối đa trên tất cả các C. Nhiều phương pháp nhận dạng có nguồn gốc từ các phương pháp tiếp cận phân loại nêu trên được sử dụng trong nhận dạng on-line như curve matching, elastic matching, neural network approach (NNA), HMMs.

2.5.1 Bắt các khúc quanh

Trong phương pháp này, các đường cong từ các ký tự được bắt và so sánh với các đường cong của các ký tự mẫu và kết quả mẫu có đường cong lớn nhất được xác định là ký tự nhận dạng được. Các đường cong phù hợp thường là một hàm của thời gian, x và các giá trị y, góc hướng của tiếp tuyến với quỹ đạo của người viết hoặc cả hai. Một cách khác là khớp mốc thời gian giữa các hệ số Fourier thu được từ x (t) và y (t) của đường cong. Phương pháp này thích hợp khi ký tự được trình bày bởi một số lượng nhỏ của các hệ số Fourier.

2.5.2 Mạng Neural

Mô phỏng hoạt động của các nơ ron thần kinh, mạng nơ ron nhân tạo là hệ thống bao gồm nhiều phần tử xử lý đơn giản (neuron) hoạt động song song. Tính năng của hệ thống này tuỳ thuộc vào cấu trúc của hệ, các trọng số liên kết nơ ron và quá trình tính toán tại các nơ ron đơn lẻ. Mạng nơ ron có thể từ dữ liệu mẫu và tổng quát hoá dựa trên các dữ liệu mẫu học [10].



Hình 2- 6 Mô hình mạng Neuron

Một nhóm các nơ ron được tổ chức theo một cách sao cho tất cả chúng đều nhận cùng một vector vào X để xử lý tại cùng một thời điểm. Việc sản sinh ra tín hiệu ra của mạng xuất hiện cùng một lúc. Vì mỗi nơ ron có một tập trọng số khác nhau nên có bao nhiêu nơ ron sẽ sản sinh ra bấy nhiêu tín hiệu ra khác nhau. Một nhóm các nơ ron như vậy được gọi là một lớp mạng. Chúng ta có thể kết hợp nhiều lớp mạng tạo ra một mạng có nhiều lớp, lớp nhận tín hiệu đầu vào (vector tín hiệu vào x) được gọi là lớp vào (input layer). Trên thực tế chúng thực hiện như một bộ đệm chứa tín hiệu đầu vào. Các tín hiệu đầu ra của mạng được sản sinh ra từ lớp ra của

mạng (output layer). Bất kỳ lớp nào nằm giữa 2 lớp mạng trên được goi là lớp ẩn (hidden layer) và nó là thành phần nội tại của mạng và không có tiếp xúc nào với môi trường bên ngoài. Số lượng lớp ẩn có thể từ 0 đến vài lớp. Mô hình nơ ron nhân tạo đòi hỏi 3 thành phần cơ bản sau [11]:

- Tập trọng số liên kết đặc trưng cho các khớp thần kinh.
- Bộ cộng (Sum) để thực hiện phép tính tổng các tích tín hiệu vào với trọng số liên kết tương ứng.
- Hàm kích hoạt (squashing function) hay hàm chuyển (transfer function) thực hiện giới hạn đầu vào của neuron.

Trong mô hình nơ ron nhân tạo mỗi nơ ron được nối với các nơ ron khác và nhận được tín hiệu xi từ chúng với các trọng số wi. Tổng thông tin vào có trọng số là: Net = $= \sum w_j x_j$.

Một ví dụ của một hệ thống nhận dạng bằng cách sử dụng ANN có thể được tìm thấy trong. Hệ thống này sử dụng thời gian trì hoãn một mạng lưới thần kinh Time delay neural network (TDNN) trong một mô-đun nhận dạng từ chữ thảo. Việc sử dụng các kỹ thuật mạng lưới công nhận dựa trên thần kinh đi qua sự cần thiết cho một bước phân khúc rõ ràng nhân vật bằng cách khai thác các đại diện thời gian của đầu vào. Đầu vào là thu được bằng cách mã hóa các quỹ đạo bút như là một chuỗi các khung hình. Mạng TDNN sau đó hoạt động trên một cửa sổ của khung hình (bao gồm một nhân vật và các bộ phận của các nước láng giềng) và sản xuất một đầu ra ở mỗi khoảng thời gian. Một ví dụ khác là hệ thống nhận dạng handprinted sử dụng các mạng thần kinh. Kiến trúc mạng được sử dụng trong hệ thống này bao gồm nhiều đại diện đầu vào, một lớp đầu tiên ẩn bằng cách sử dụng các lĩnh vực tiếp nhận, đầy đủ kết nối lớp thứ hai ẩn và một lớp đầu ra cuối cùng được chia sẻ đầy đủ kết nối. Các dữ liệu đầu vào bao gồm thông tin về tính năng nét vẽ và hình ảnh của ký tự.

2.5.3 Mô hình Markov ẩn

Đây là một phương pháp thống kê dựa trên nguyên tắc phân loại bởi khả năng tối đa. HMMs là một phương pháp thường được sử dụng trong cả hai nhận dạng

chữ viết tay off-line và on-line... chi tiết về phương pháp này sẽ được trình bày trong chương sau.

2.6 Hậu Xử lý

Mục tiêu của hậu xử lý để sử dụng thêm thông tin theo ngữ cảnh để nhận dạng chữ viết, mục đích là tăng tính chính xác của hệ thống nhận dạng trong trường hợp ký tự cần nhận dạng có sự nhập nhằng ví dụ chữ o và số 0, chữ U và V…. Hậu xử lý có thể là một số thống kê về những sai lầm nhận dạng phổ biến để ra quyết định xóa bỏ hay thay thế một ký tự nhận dạng để phù hợp với ngôn ngữ tự nhiên. Một kỹ thuật xử lý sau cao cấp hơn là sử dụng một mô hình ngôn ngữ, chẳng hạn như ngữ pháp, để xác định các hạn chế về bối cảnh trong đó từ hoặc ký tự được sử dụng. Một mô hình ngôn ngữ có thể là xác suất, liên quan đến số liệu thống kê về xác suất của một từ khác từ (bigrams) hoặc n xảy ra theo tuần tự (ngrams). Chuỗi Markov rất thích hợp cho các loại ngôn ngữ mô hình này.

2.7 Tóm tắt thông tin

Trong chương này, chúng ta đã tập trung tìm hiểu chi tiết các khía cạnh, các chức năng của một hệ thống dạng chữ viết tay trực tuyến bao gồm cả vấn đề nhận dạng, các thiết bị đầu vào, tiền xử lý, khai thác tính năng, phương pháp nhận dạng và xử lý sau.. Tiền xử lý sẽ giúp loại bỏ thông tin không liên quan và giữ các thông tin hữu ích từ các dữ liệu thô chụp bởi thiết bị đầu vào trong luận văn này tác giả sử dụng đối tượng picturebox của visual Studio. Bước bao gồm các bước nhỏ như làm mịn, phân đoạn và lấy mẫu để đưa vào bước nhận dạng. Chương này cũng giới thiệu một số kỹ thuật nhận dạng như mạng ANN, phương pháp, bắt khúc quanh …Chương kế tiếp tác giả sẽ giới thiệu chi tiết về phương pháp nhận dạng sử dụng mô hình Markov ẩn.

CHƯƠNG 3: GIỚI THIỆU MÔ HÌNH MARKOV

Được giới thiệu và nghiên cứu vào cuối những năm 1960, phương pháp thống kê dựa trên lý thuyết Markov hay mô hình Markov ẩn đã ngày càng trở nên phổ biến trong những năm gần đây. Có hai lý do lý giải cho việc trên. Thứ nhất mô hình rất phong phú trong cấu trúc toán học và do đó có thể hình thành cơ sở lý thuyết để sử dụng trong một loạt các ứng dụng.Thứ hai các mô hình, khi áp dụng đúng cách, vận dụng rất tốt trong thực tế đối với các ứng dụng quan trọng. Trong chương này chúng ta sẽ tìm hiểu sâu về mô hình Markov ẩn, các vấn đề cơ bản của mô hình HMM, các giải thuật để giải quyết các bài toán đó. Giới thiệu tổng quang mô hình hệ thống nhận dạng chữ viết sử dụng thuật toán Markov. Cơ sở lý thuyết trong chương này dựa trên các lý thuyết do Lawrence Rabiner và Juang [2]

3.1Giới thiệu

Các kết quả đầu ra của quá trình xử lý thực tế có thể được mô tả như là tín hiệu. Các tín hiệu có thể được rời rạc hoá trong tự nhiên (ví dụ: các ký tự từ một bảng chữ cái, lượng tử hóa vector từ codebook,…), hoặc liên tục trong tự nhiên (ví dụ, giọng nói mẫu, đo lường nhiệt độ, âm nhạc, …). Các nguồn tín hiệu có thể không đổi (tính chất thống kê của nó không thay đổi theo thời gian), hoặc thay đổi (các đặc tính tín hiệu thay đổi theo thời gian). Các tín hiệu có thể không có tạp âm (phát từ một nguồn duy nhất), hoặc có thể bị hỏng bởi các nguồn tín hiệu khác (ví dụ:, tiếng ồn) hoặc bằng cách biến dạng truyền, vang, …

Một vấn đề được quan tâm cơ bản là đặc điểm tín hiệu trong thế giới thực của các mô hình tín hiệu. Đó là lý do tại sao chúng ta quan tâm đến việc mô hình hóa tín hiệu. Trước hết, một mô hình tín hiệu có thể cung cấp cơ sở cho một mô tả lý thuyết của một hệ thống xử lý tín hiệu, có thể dùng để xử lý tín hiệu để cung cấp một đầu ra mong muốn. Ví dụ, nếu chúng ta quan tâm trong việc tăng cường một tín hiệu âm thanh bị hỏng bởi tiếng ồn và sự biến dạng truyền dẫn, chúng ta có thể sử dụng mô hình tín hiệu để thiết kế một hệ thống tối ưu sẽ loại bỏ các tiếng ồn và quay lại sự biến dạng truyền dẫn. Lý do thứ hai lý giải tại sao mô hình tín hiệu rất quan trọng là chúng có khả năng có khả năng cho phép chúng ta tìm hiểu về các nguồn tín hiệu. (tức là, quá trình thế giới thực mà sản xuất ra các tín hiệu) mà không cần phải có

nguồn sẵn có. Thuộc tính này là đặc biệt quan trọng khi các đòi hỏi để nhận được tín hiệu từ nguồn thực tế là khó khăn. Trong trường hợp này, với một mô hình tín hiệu tốt, chúng ta có thể mô phỏng các nguồn và tìm hiểu càng nhiều càng tốt thông qua mô phỏng. Cuối cùng, lý do quan trọng nhất lý giải tại sao mô hình tín hiệu rất quan trọng là chúng thường làm việc rất tốt trong thực tế, và cho phép chúng ta nhận ra hệ thống thực tế rất quan trọng, ví dụ như hệ thống dự báo, hệ thống nhận dạng, xác định hệ thống, …, một cách rất hiệu quả.

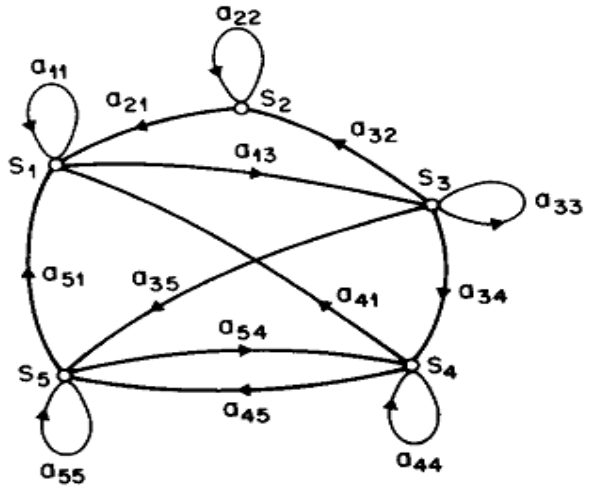
Đây là một số lựa chọn tốt cho những gì của mô hình tín hiệu được sử dụng để mô tả các thuộc tính của một tín hiệu đã cho. Nói rộng ra người ta có thể phân chia các loại mô hình tín hiệu vào lớp các mô hình xác định, và lớp các mô hình thống kê. Mô hình xác định khai thác một số đặc tính cụ thể được biết đến của các tín hiệu, ví dụ như, tín hiệu là một làn sóng sin, hoặc một tổng của các hàm mũ, … Trong những trường hợp này, đặc điểm kỹ thuật của mô hình tín hiệu nói chung là đơn; tất cả những gì là cần thiết để xác định (ước tính) giá trị của các tham số của mô hình tín hiệu (ví dụ, biên độ, tần số, giai đoạn của một hàm sóng sin, biên độ và tỷ lệ của hàm mũ, …). Lớp lớn thứ hai của các mô hình tín hiệu là tập hợp của các mô hình thống kê, trong đó cố gắng để mô tả các tính chất thống kê của tín hiệu. Ví dụ các mô hình thống kê như vậy bao gồm các quy trình Gaussbian, quá trình Poisson, quá trình Markov, và các quá trình Markov ẩn, một số khác. Các giả định cơ bản của mô hình thống kê là tín hiệu có thể được mô tả như là một quá trình với tham số ngẫu nhiên, và các tham số của quá trình ngẫu nhiên có thể được xác định (ước tính) với độ chính xác được xác định rõ. Đối với các ứng dụng được quan tâm, cụ thể là xử lý tiếng nói, cả hai mô hình tín hiệu xác định và ngẫu nhiên đã thành công tốt đẹp. Trong chương này, chúng ta sẽ quan tâm đến một loại mô hình tín hiệu ngẫu nhiên, cụ thể là mô hình Markov ẩn (HMM). Đầu tiên chúng ta sẽ xem xét các lý thuyết về chuỗi Markov và sau đó mở rộng các ý tưởng về các lớp của mô hình Markov ẩn bằng cách sử dụng một số ví dụ đơn giản. Sau đó ta sẽ tập trung sự chú ý về vấn đề cơ bản cho thiết kế HMM, cụ thể là: đánh giá xác suất (khả năng) của một chuỗi các quan sát cho HMM cụ thể, xác định một trình tự tốt nhất của mô

hình trạng thái, và điều chỉnhcác thông số mô hình để tính toán tốt nhất cho các tín hiệu quan sát được.

3.2 Khái niệm về mô hình

Mô hình Markov ẩn là mô hình thống kê trong đó hệ thống được mô hình hóa được cho là một quá trình Markov với các tham số không biết trước và nhiệm vụ là xác định các tham số ẩn từ các tham số quan sát được. Các tham số của mô hình được rút ra sau đó có thể được sử dụng để thực hiện các phân tích kế tiếp, ví dụ ứng dụng cho nhận dạng mẫu.

Hãy xem xét một hệ thống có thể được mô tả ở bất kỳ thời điểm nào là một phần tử trong một tập hợp N trạng thái khác biệt, S_1, S_2. . ., S_n, như minh họa trong hình 3-1 (trong đó N = 5 cho đơn giản). Tại những khoảng thời gian rời rạc cách nhau đều đặn, hệ thống trải qua sự thay đổi của trạng thái (có thể trở lại tình trạng tương tự) theo một tập hợp xác suất liên kết với trạng thái. Ta biểu thị thời gian tức thời liên quan đến thay đổi trạng thái t = 1, 2,. . . , Và chúng ta biểu thị tình trạng thực tế tại thời điểm t như q_t. Một mô tả xác suất đầy đủ của hệ thống trên, nói chung, yêu cầu đặc điểm kỹ thuật của trạng thái hiện hành (tại thời điểm t), cũng như tất cả các trạng thái trước đó.



Hình 3- 1 Một chuỗi Markov với 5 trạng thái với các chuyển trạng thái

Đối với trường hợp đặc biệt của một trật tự rời rạc đầu tiên, chuỗi Markov, xác suất mô tả bị cắt bớt chỉ là hiện tại và trạng thái trước, tức là,

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \cdots] = P[q_t = S_j | q_{t-1} = S_i]. \quad (3\text{-}1)$$

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \le i, j \le N \quad (3\text{-}2)$$

$$a_{ij} \ge 0$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad (3\text{-}3)$$

Hơn nữa chúng ta chỉ xem các quy trình trong đó phía bên phải của (3-1) là độc lập với thời gian, từ đó dẫn đến các tập hợp các xác suất chuyển trạng thái aij của dưới dạng với hệ số chuyển đổi trạng thái có thuộc tính vì chúng tuân theo tiêu chuẩn ràng buộc ngẫu nhiên. Quá trình ngẫu nhiên ở có thể được gọi là một mô hình Markov quan sát được kể từ khi đầu ra của quá trình là tập hợp của các trạng thái trong từng khoảnh khắc của thời gian, nơi mà mỗi trạng thái tương ứng với một

sự kiện vật lý (quan sát).Để thiết lập những ý tưởng, hãy xem xét một mô hình Markov 3 trạng thái đơn giản của thời tiết.Chúng ta giả định rằng một lần một ngày (ví dụ, vào buổi trưa), thời tiết được quan sát là một trong những việc sau:

trạng thái 1: mưa (tuyết)

trạng thái 2: mây

trạng thái 3: nắng.

Chúng ta giả định rằng thời tiết trong ngày đặc trưng bởi một trong ba trạng thái trên, và ma trận xác suất chuyển trạng thái:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

Cho rằng thời tiết vào ngày 1 (t = 1) là nắng (trạng thái 3), chúng ta có thể đặt câu hỏi: Xác suất (theo mô hình) mà thời tiết trong 7 ngày tới sẽ là "nắng nắng mưa mưa-nắng-mây-nắng …" là gì? Nói chính thức hơn, chúng ta xác định trình tự quan sát O là O = {S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3} tương ứng với t = 1, 2,. . ., 8 và chúng ta muốn xác định xác suất của O, cho mô hình này. Xác suất này có thể được thể hiện và đánh giá như sau:

$$
\begin{aligned}
P(O|\text{Model}) &= P[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3|\text{Model}] \\
&= P[S_3] \cdot P[S_3|S_3] \cdot P[S_3|S_3] \cdot P[S_1|S_3] \\
&\quad \cdot P[S_1|S_1] \cdot P[S_3|S_1] \cdot P[S_2|S_3] \cdot P[S_3|S_2] \\
&= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\
&= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned} \tag{3-4}
$$

các ký hiệu

$$
\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N \tag{3-5}
$$

để biểu thị xác suất trạng thái ban đầu.

Một câu hỏi thú vị chúng ta có thể hỏi (và trả lời bằng cách sử dụng các mô hình): Cho rằng mô hình này là trong một trạng thái biết đến, xác suất nó vẫn ở trạng thái đó cho ngày chính xác d là gì? Xác suất này có thể được đánh giá là xác suất của các quan sát chuỗi.

$$
O = \{\underset{1}{S_i}, \underset{2}{S_i}, \underset{3}{S_i}, \cdots, \underset{d}{S_i}, \underset{d+1}{S_j} \neq S_i\}, \tag{3-6}
$$

đưa ra mô hình, đó là

$$
P(O|\text{Model}, q_1 = S_i) = (a_{ii})^{d-1}(1 - a_{ii}) = p_i(d). \tag{3-7}
$$

Pi số lượng (d) (rời rạc) xác suất mật độ hàm thời gian d ở trạng thái i.Mật độ thời gian theo cấp số nhân là đặc trưng của thời gian trạng thát một chuỗi Markov. Dựa trên pi (d), chúng ta có thể dễ dàng tính toán số lượng dự kiến của các quan sát (thời gian) trong một trạng thái, điều kiện bắt đầu trong trạng thái đó như:

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d)$$

$$= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) = \frac{1}{1 - a_{ii}}. \quad (3\text{-}8)$$

Vì vậy, số ngày nắng liên tục dự kiến, theo mô hình, là 1/(0,2) = 5, cho mây nó là 2,5 mưa là 1,67.

Mở rộng Mô hình Markov ẩn

Đến nay chúng ta đã xem xét mô hình Markov, trong đó mỗi trạng thái tương ứng với một sự kiện (vật lý) quan sát được. Mô hình này là quá hạn chế để áp dụng đối với nhiều vấn đề quan tâm. Trong phần này chúng ta mở rộng khái niệm về mô hình Markov để bao gồm các trường hợp quan sát là một chức năng xác suất của các trạng thái-có nghĩa là, mô hình (gọi là một mô hình Markov ẩn) là một quá trình ngẫu nhiên gấp đôi nhúng với một quá trình ngẫu nhiên cơ bản rằng không quan sát được (ẩn), nhưng chỉ có thể được quan sát thông qua một tập hợp các quy trình ngẫu nhiên tạo ra chuỗi các quan sát. Để khắc phục những ý tưởng, xem xét các mô hình sau đây của một thí nghiệm ném đồng xu đơn giản.

Mô hình ném đồng xu: Giả sử kịch bản sau đây. Bạn đang ở trong một căn phòng với một rào cản (ví dụ, một bức màn) thông qua đó bạn không thể nhìn thấy những gì đang xảy ra. Ở phía bên kia của rào cản là một người đang thực hiện ném một đồng xu (hoặc nhiều đồng xu) thử nghiệm. Người đó sẽ không cho bạn biết bất cứ điều gì về những gì ông đang làm chính xác, anh ta sẽ chỉ cho bạn biết kết quả của mỗi đồng xu được ném. Như vậy, một chuỗi các thí nghiệm tung đồng xu bị ẩn được thực hiện, với chuỗi quan sát bao gồm một loạt của đầu và đuôi, ví dụ, một chuỗi quan trắc điển hình trong đó H là viết tắt của đầu và F là viết tắt của đuôi.

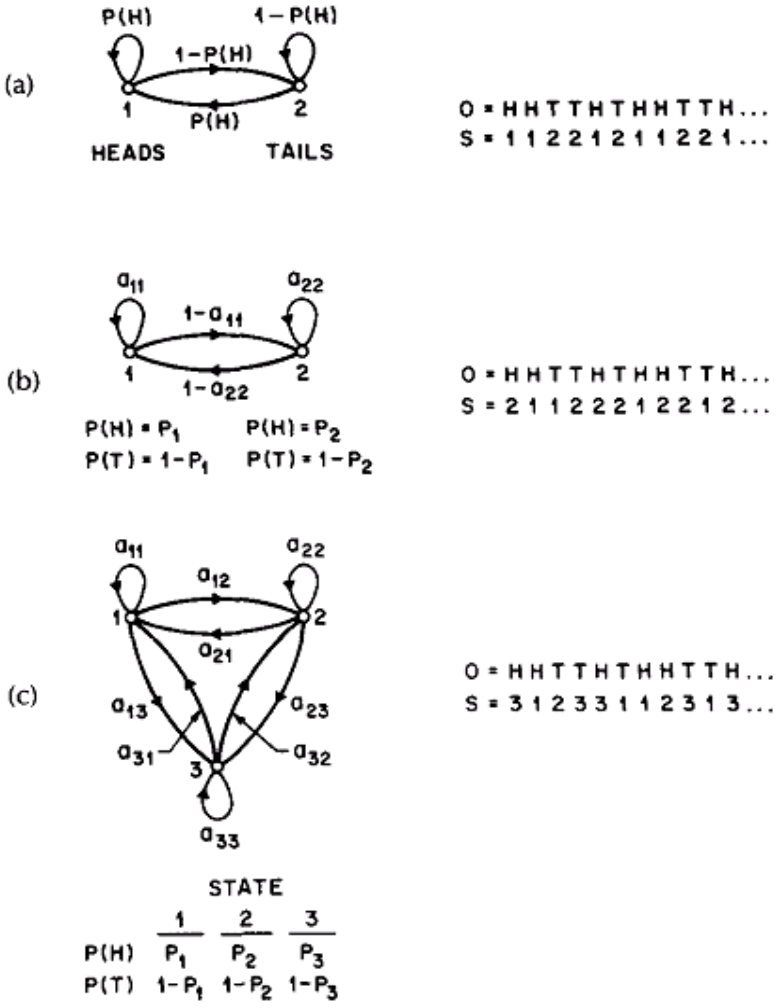
$$\mathbf{O} = O_1 O_2 O_3 \cdots O_T$$

$$= \mathcal{H}\ \mathcal{H}\ \mathcal{T}\ \mathcal{T}\ \mathcal{T}\ \mathcal{H}\ \mathcal{T}\ \mathcal{T}\ \mathcal{H} \cdots \mathcal{H}$$

Với kịch bản ở trên, vấn đề quan tâm là làm thế nào để chúng ta xây dựng một HMM để giải thích (mô hình) chuỗi quan sát của đầu và đuôi. Vấn đề đầu tiên một phải đối mặt là quyết định những gì các trạng thái trong mô hình tương ứng, và sau đó quyết định như thế nào nhiều trạng thái phải có trong mô hình. Một sự lựa chọn có thể sẽ có giả định rằng chỉ có một đồng xu đã bị tung sai lệch. Trong trường hợp này, chúng ta có thể mô hình hóa với một mô hình 2- trạng thái, nơi mà mỗi trạng thái tương ứng với một mặt của đồng xu (tức là, sấp hoặc ngửa).Mô hình này được mô tả trong hình. 3-2 (a). Trong trường hợp này mô hình Markov là quan sát được, và vấn đề duy nhất cho các đặc điểm kỹ thuật đầy đủ của mô hình sẽ có quyết định giá trị tốt nhất dẫn đến sai số. Thật thú vị, một HMM tương đương của hình. 3-2 (a) sẽ là một thoái hóa I- trạng thái mô hình, nơi trạng thái tương ứng với đồng xu sai lệch duy nhất, và các tham số không rõ là sai số của đồng xu.

Một hình thức thứ hai của HMM để giải thích các trình tự quan sát của đồng xu ném kết quả được đưa ra trong hình. 3-2 (b).Trong trường hợp này có 2 trạng thái trong mô hình và mỗi trạng thái tương ứng với một sai lệch, khác nhau, đồng xu được ném. Mỗi trạng thái được đặc trưng bởi một phân bố xác suất sấp và ngửa, và quá trình chuyển đổi giữa các trạng thái được đặc trưng bởi một ma trận chuyển đổi trạng thái. Cơ chế vật lý mà tài khoản để chuyển trạng thái được lựa chọn chính có thể là tập hợp đồng xu tung độc lập, hoặc một số sự kiện xác suất khác.

Một hình thức thứ của HMM để giải thích các trình quan của một đồng xu ném kết quả được đưa ra trong hình. 3-2 (c).Mô hình này tương ứng với sử dụng 3 tiền xu thiên lệch, và lựa chọn trong số cả ba, dựa trên một số trường xác suất.

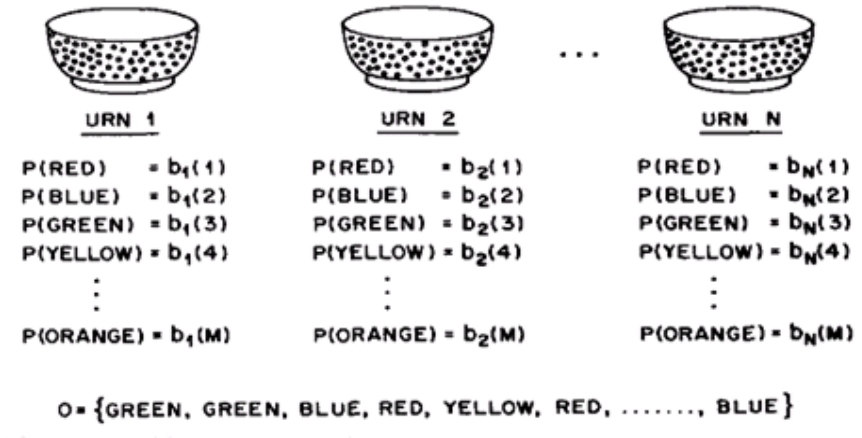


Hình 3- 2 ba mô hình Markov tương ứng với việc tung 1,2,3 đồng xu

Với sự lựa chọn giữa ba mô được trong hình.2 để giải thích các trình quan sát của đầu và đuôi, một câu hỏi tự nhiên sẽ là mô hình tốt nhất phù hợp với quan sát thực tế. Nó sẽ được rõ ràng rằng 1-xu mô hình đơn giản của hình. 3-2 (a) chỉ có 1 tham số chưa biết, mô hình 2 đồng xu của hình.3-2 (b) có 4 thông số chưa được biết đến, và mô hình 3-xu của hình. 3-2 (c) có 9 thông số chưa biết. Như vậy, với mức độ cao hơn của sự tự do, HMMs lớn hơn dường như vốn đã được nhiều hơn khả năng của mô hình một loạt các thí nghiệm tung đồng xu sẽ tương đương nhỏ hơn mô hình. Mặc dù điều này là đúng về mặt lý thuyết, chúng ta sẽ thấy sau này trong bài báo này xem xét thực tế áp đặt một số hạn chế mạnh mẽ vào kích thước của các

mô hình mà chúng ta có thể xem xét.Hơn nữa, nó chỉ có thể là trường hợp mà chỉ một đồng xu được ném. Sau đó, bằng cách sử dụng mô hình 3-xu Hình. 3-2 (c) không phù hợp, kể từ khi sự kiện vật lý thực tế sẽ không tương ứng với mô hình đang được sử dụng tức là, chúng ta sẽ sử dụng theo hệ thống quy định.

Mô hình chiếc bình và bóng: Để mở rộng những ý tưởng của HMM đến một tình hình phức tạp hơn đôi chút, hãy xem xét chiếc bình và hệ thống quả bóng tong hình 3-3. Chúng ta cho rằng có N (rất lớn) bình thủy tinh trong một phòng.Trong mỗi chiếc bình từng có một số lượng lớn các quả bóng màu.Chúng ta giả sử có M màu sắc riêng biệt của các quả bóng. Quá trình vật lý để có được quan sát như sau. Một vị thần trong phòng, và theo một số quá trình ngẫu nhiên, ông (hoặc bà) chọn chiếc bình ban đầu.Từ chiếc bình này, một quả bóng được chọn ngẫu nhiên, và màu sắc của nó được ghi nhận là quan sát. Quả bóng sau đó được thay thế trong chiếc bình mà từ đó nó đã được lựa chọn. Chiếc bình mới sau đó lựa chọn theo quá trình lựa chọn ngẫu nhiên gắn liền với chiếc bình hiện tại, và quá trình lựa chọn quả bóng là lặp đi lặp lại. Toàn bộ quá trình này tạo ra một chuỗi quan sát hữu hạn của màu sắc, mà chúng ta muốn mô hình là đầu ra quan sát của HMM.



Hình 3- 3 Mô hình N- trạng thái minh họa các các model HMM rời rạc

Nó sẽ chỉ ra được rõ ràng là HMM đơn giản tương ứng với chiếc bình và quả bóng là một trong đó mỗi trạng thái tương ứng với một chiếc bình cụ thể, và xác

suất (quả bóng) màu được định nghĩa cho mỗi trạng thái.Sự lựa chọn của bình được quyết định bởi các ma trận chuyển đổi trạng thái của HMM.

Các yếu tố của HMM

Các ví dụ trên cho chúng ta một ý tưởng khá tốt về những gì là một HMM và làm thế nào nó có thể được áp dụng cho một số kịch bản đơn giản.Bây giờ chúng ta chính thức xác định các yếu tố của một HMM, và giải thích làm thế nào mô hình tạo ra các trình tự quan sát.

Một HMM được đặc trưng bởi những yếu tố sau:

- N, số lượng của các trạng thái trong mô hình.Mặc dù các trạng thái là ẩn, đối với nhiều ứng thực tế thường là một số ý nghĩa vật lý thuộc các trạng thái hoặc bộ trạng thái của mô hình. Do đó, trong các thí nghiệm tung đồng xu, mỗi trạng thái tương ứng với một đồng xu thiên lệch khác biệt. Trong mô hình chiếc bình và quả bóng, các trạng thái tương ứng với các bình. Nói chung các trạng thái đang kết nối với nhau theo một cách mà bất kỳ trạng thái nào có thể đạt được từ bất kỳ trạng thái khác (ví dụ, một mô hình ergodic), tuy nhiên, chúng ta sẽ thấy sau này trong giấy khác có thể có mối liên kết của các trạng thái thường quan tâm. Ta biểu thị các trạng thái riêng lẻ là $S = \{S_1, S_2, . . ., S_n\}$, và trạng thái tại thời điểm t là q_t.
- M, số lượng các biểu tượng quan sát khác biệt cho mỗi trạng thái, tức là, kích thước bảng chữ riêng biệt.Các biểu tượng quan sát tương ứng với đầu ra vật lý của hệ thống đang được mô hình.Đối với những thí nghiệm ném đồng xu những biểu tượng quan sát được chỉ đơn giản là đứng đầu hoặc cuối, bóng và mô hình chiếc bình chúng đã được các màu sắc của các quả bóng được lựa chọn từ bình. Ta biểu thị các biểu tượng riêng lẻ là $V = \{v_1, v_2,\ldots,v_m\}$.
- Xác suất chuyển đổi trạng thái phân phối A = {a,} nơi

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \le i, j \le N. \tag{3-8}$$

Đối với trường hợp đặc biệt nơi mà bất cứ nhà nước có thể đạt được bất kỳ trạng thái khác trong một bước duy nhất, chúng ta có $a_{ij} > 0$ cho tất cả i j. Đối với các loại khác của HMMs, chúng ta sẽ có $a_{ij} = 0$ đối với một hoặc nhiều cặp (i, j)

➢biểu tượng quan sát xác suất phân phối ở trạng thái j, $B = \{b_j(k)\}$,

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad 1 \le j \le N$$
$$1 \le k \le M. \tag{3-9}$$

➢Sự phân bố trạng thái ban đầu $\pi = \{\pi_i\}$

$$\pi_i = P[q_1 = S_i], \quad 1 \le i \le N. \tag{3-10}$$

Với giá trị thích hợp của N, M, A, B, và π, HMM có thể được sử dụng như bộ sinh để cung cấp một chuỗi quan sát:

$$O = O_1 O_2 \cdots O_T \tag{3-11}$$

(mà mỗi quan sát O, là một trong các biểu tượng từ V, và T là số lượng quan sát theo thứ tự) như sau:

(1) Chọn một trạng thái ban đầu $q_1 = S_i$ theo phân phối trạng thái ban đầu T.

2) Đặt t = 1.

3) $O_t = v_k$ theo phân bố xác suất biểu tượng trong S_I trạng thái, tức là, b_i (k).

4) Chuyển trạng thái mới $q_{t+1} = S_j$ theo phân phối trạng thái chuyển xác suất cho trạng thái S, nghĩa là một.

5) Đặt t = t + 1; trở lại bước 3) nếu t <T, nếu không kết thúc thủ tục.

Thủ tục trên có thể được sử dụng như bộ sinh của các quan sát, và như một mô hình cho một trình tự quan sát được tạo ra bởi HMM thích hợp.

Nó có thể được nhìn thấy từ thảo luận ở trên là một đặc điểm kỹ thuật hoàn chỉnh của một HMM yêu cầu đặc điểm kỹ thuật của hai thông số mô hình (N và M), đặc điểm kỹ thuật của các biểu tượng quan sát, và đặc điểm kỹ thuật của xác suất ba biện pháp A, B, và T. Để thuận tiện,ta sử dụng các ký hiệu:

$$\lambda = (A, B, \pi) \quad (3\text{-}12)$$

để chỉ ra các thông số thiết lập của mô hình.

Quan sát không tốn bộ nhớ, nghĩa là chuỗi các quan sát có xác suất chỉ phụ thuộc vào trạng thái ngay trước đó (nên không cần lưu bộ nhớ nhiều).

3.3Ba vấn đề cơ bản của HMM

Với hình thức của HMM trong những phần trước, có ba vấn đề cơ bản cần quan tâm phải được giải quyết cho mô hình hữu ích trong ứng dụng thế giới thực.Những vấn đề này như sau:

Vấn đề 1: Với trình tự quan sát $O = O_1\ O_2 \ldots O_T$ Hoặc, và một mô hình $\lambda = (A, B, \pi)$, làm thế nào để chúng có hiệu quả tính toán $P(O|\lambda)$, xác suất của chuỗi quan sát, cho các mô hình?

Vấn đề 2: Do trình tự quan sát $O = O_1\ O_2\ O_T$, và mô hình A, làm thế nào để chúng ta chọn một chuỗi trạng thái tương ứng $Q = q_1\ q_2\ q_j$ trong đó là tối ưu theo một nghĩa (ví dụ, giải thích tốt nhất cho quan sát s)?

Vấn đề 3: Làm thế nào để chúng ta điều chỉnh các thông số mô hình $A = (A, B, \pi)$ để tối đa hóa $P(O|\lambda)$?

Chi tiết từng vấn đề

Vấn đề 1 là vấn đề đánh giá, cụ thể là một mô hình và và trình tự quan sát, làm thế nào để chúng ta tính xác suất rằng trình tự quan sát được tạo ra bởi mô hình. Chúng ta cũng có thể xem vấn đề là một trong những đánh giá như thế nào cho một mô hình đưa ra phù hợp với một chuỗi quan sát đưa ra. Quan điểm thứ hai là cực kỳ hữu ích. Ví dụ, nếu chúng ta xem xét trường hợp trong đó chúng ta đang cố gắng để lựa chọn giữa một số mô hình cạnh tranh, giải pháp cho vấn đề 1 cho phép chúng ta lựa chọn mô hình phù hợp nhất với các quan sát.

Vấn đề 2 là một trong những vấn đề mà chúng ta cố gắng khám phá ra bộ phận ẩn của các mô hình, ví dụ, để tìm chuỗi trạng thái "chính xác".Nó sẽ được xóa cho tất cả các trường hợp thoái hóa mô hình, đó là không có trạng thái"đúng" trình tự được tìm thấy. Do đó cho các tình huống thực tế, chúng ta thường sử dụng một tiêu chí tối ưu để giải quyết vấn đề này một cách tốt nhất có thể.Thật không may, chúng ta sẽ thấy, có một số tiêu chí tối ưu hợp lý có thể được áp đặt, và do đó sự lựa chọn của tiêu chuẩn là một chức năng mạnh mẽ của mục đích sử dụng cho chuỗi trạng thái đã phát hiện.Tiêu biểu dùng có thể được để tìm hiểu về cấu trúc của mô hình, để tìm chuỗi trạng thái tối ưu cho nhận dạng giọng nói liên tục, hoặc để có được số liệu thống kê trung bình của các trạng thái cá nhân,...

Vấn đề 3 là một trong những vấn đề mà chúng ta cố gắng tối ưu hóa các thông số mô hình tốt nhất mô tả một chuỗi quan sát cho đi kèm khoảng.Trình tự quan sát được sử dụng để điều chỉnh các thông số mô hình được gọi là một trình tự đào tạo kể từ khi nó được sử dụng để "đào tạo" HMM.Vấn đề đào tạo là một trong những vấn đề quan trọng đối với hầu hết ứng dụng của HMMs, vì nó cho phép chúng ta cho thích ứng tối ưu thông số mô hình đào tạo quan sát dữ liệu, tức là, để tạo ra mô hình tốt cho hiện tượng thực tế.

Để sửa chữa những ý tưởng, hãy xem xét bên nhận từ sau đây đơn giản bị cô lập phát biểu. Đối với mỗi từ của một từ vựng từ W, chúng ta muốn thiết kế một HMM N-trạng thái riêng biệt.. Chúng ta đại diện các tín hiệu lời nói của một từ được đưa ra như là một chuỗi thời gian của vector quang phổ được mã hóa. Chúng

ta giả định rằng mã hóa được thực hiện bằng cách sử dụng một danh bạ quang phổ với vector M quang phổ duy nhất, vì vậy mỗi quan sát là chỉ số của các vector phổ gần nhất (trong một số ý nghĩa quang phổ) để các tín hiệu lời nói ban đầu. Như vậy, đối với mỗi từ vựng, chúng ta có một trình tự huấn luyện bao gồm một số lần lặp lại trình tự của các chỉ số danh bạ của từ này (một hoặc nhiều người nói). Nhiệm vụ đầu tiên là xây dựng mô hình từ cá nhân. Nhiệm vụ này được thực hiện bằng cách sử dụng các giải pháp cho Vấn đề 3 để ước tính tối ưu các thông số mô hình cho mỗi mô hình từ ngữ. Để phát triển một sự hiểu biết về ý nghĩa vật lý của các trạng thái mô hình, chúng ta sử dụng giải pháp cho Vấn đề 2 phân đoạn của các trình tự huấn luyện từ ngữ vào trạng thái, và sau đó nghiên các thuộc tính của các vectơ quang phổ dẫn đến các quan sát xảy ra trong mỗi trạng thái. Mục tiêu ở đây sẽ được thực hiện cải tiến về mô hình (nhiều trạng thái, danh bạ kích thước khác, …) để cải khả năng mô hình hóa các trình tự từ nói. Cuối cùng, một khi các thiết lập của W HMMs đã được thiết kế và tối ưu hóa và nghiên cứu kỹ lưỡng, công nhận của một từ không được thực hiện bằng cách sử dụng giải pháp cho vấn đề 1 số điểm từ ngữ mỗi mô hình dựa trên trình tự quan sát được đưa ra thử nghiệm, và chọn từ có số điểm mô hình là cao nhất (tức là khả năng cao nhất).

3.4 Giải quyết ba vấn đề của HMM

3.3.1 Thuật toán tiến – thuật toán lùi:

Toán tử tiến αt(t) là xác suất chuỗi quan sát từng phần tiến đến thời điểm t và trạng thái si ở thời điểm t với điều kiện mô hình đã cho:

$$\alpha t\,(i)= P(o1o2\ ...ot, qt = si\ |\lambda) \quad (3\text{-}13)$$

Dễ dàng thấy rằng:

$$\alpha 1(i)= \pi ibi\,(o1),\ 1 \le i \le N$$

$$P(O|\lambda)= \sum_{i=1}^{N} \alpha_T\,(i) \quad (3\text{-}14)$$

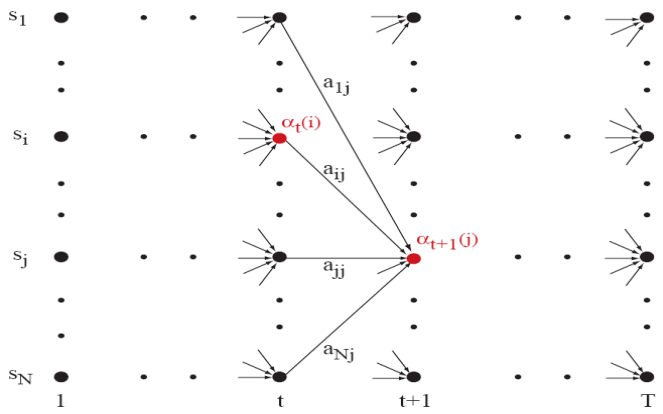
Theo phương pháp quy nạp

$$\alpha t+1\ (j)=[\sum_{i=1}^{N} \alpha_t\ (i)\ a_{ij}\ ]\ bj\ (ot+1),\ 1\leq t\leq T-1,\ 1\leq j\leq N \quad (3\text{-}15)$$

Số phép tính: N2T.

Ví dụ: N=5, T=100 ➔ 52.100 phép tính,(thay vì 1072)

thuật toán tiến



Hình 3- 4 Mô hình thuật toán tiến

Thuật toán lùi:

Tương tự để xác định toán tử lùi, βt(i), khi khả năng xảy ra của chuỗi quan sát cục bộ từ thời điểm t+1 đến kết thúc, biết trước trạng thái si ở thời điểm t và với điều kiện mô hình đã cho

$$\beta t(i)= P(ot+1ot+2...oT\ |qt = si,\lambda) \quad (3\text{-}16)$$

Có thể dễ dàng nhận ra rằng

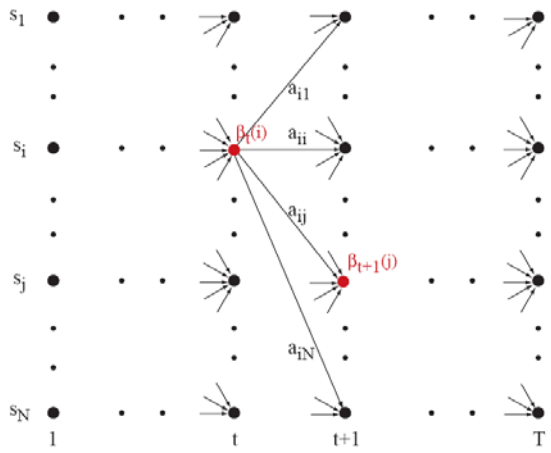
$$\beta T(i)=1,\ 1\leq i\leq N$$

$$\text{và } P(O|\lambda)=\sum_{i=1}^{N} \Pi_i b_i(o_1)\beta_1(i) \quad (3\text{-}17)$$

Theo phương pháp quy nạp

$$\beta t(i)=\sum_{j=1}^{N} a_{ij} b_j(o_{t+1})\beta_{t+1}(j) \quad (t=T-1,T-2,...,1;\ 1 \le i \le N) \text{ (3-10)}$$

Diễn tả thủ tục lùi:



Hình 3- 5 Mô tả thuật toán lùi

Tìm chuỗi trạng thái tối ưu:

+ Một tiêu chuẩn để lựa chọn trạng thái tối ưu qt là cực đại hóa số trạng thái đúng.

+ Toán tử $\gamma_t(i)$ là xác suất của hệ thống ở trạng thái si tại thời điểm t, với điều kiện cho chuỗi quan sát O và mô hình λ đã cho:

$$\gamma(i) = P(q_t = s_i \mid O, \lambda) \sum_{t=1}^{N} \gamma_t(i) = 1 \quad , \forall t \text{ (3-18)}$$

+ Chú ý rằng nó có thể biểu diễn dưới dạng sau $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)}$

+ Tuy nhiên với tiêu chuẩn tối ưu riêng phần thì xảy ra vấn đề là chuỗi trạng thái tối ưu có thể không tuân theo những ràng buộc chuyển tiếp trạng thái.

+ Một tiêu chuẩn tối ưu khác là cực đại hóa P(Q,O|λ). Điều này có thể tìm thấy bằng thuật toán Viterbi.

+ Với $\delta_t(i)$ là xác suất xảy ra cao nhất trên một đường dẫn tính với t lần quan sát đầu tiên:

$$\delta_t(i) = \max_{q_1,q_2,...,q_{t-1}} P(q_1, q_2,..., q_{t-1}, q_t = s_1, o_1 o_2 ... o_t \mid \lambda) \quad (3\text{-}19)$$

+ Theo phương pháp quy nạp:

$$\delta_{t+1}(i) = [\max_i \delta_t(i) a_{ij}] b_i(o_{t+1}) \quad (3\text{-}20)$$

+ Để thu được chuỗi trạng thái, ta cần theo dõi chuỗi trạng thái mà cho đường dẫn tốt nhất ở thời điểm t đến trạng thái si. Chúng ta thực hiện điều này trong một mảng $\psi_t(i)$.

3.2.2 Thuật toán Viterbi

+ Khởi đầu:

$$\delta_1(i) = \pi_i b_i(o_i)$$

$$\psi_1(i) = 0 \quad (3\text{-}21)$$

+ Đệ quy:

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a i_j] b_j(o_t), 2 \le t \le T$$

$$\psi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], 2 \le t \le T \quad (3\text{-}22)$$

+ Kết thúc:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)]$$

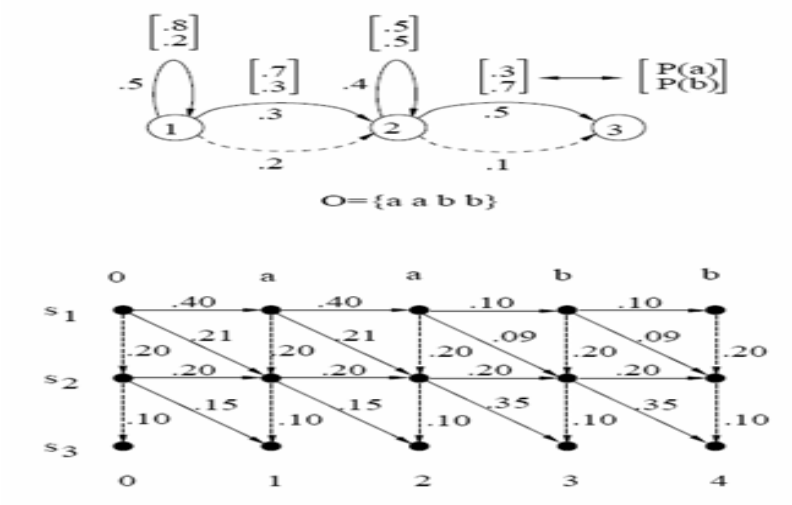
$$q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)] \quad (3\text{-}23)$$

+ Quay lui tìm đường dẫn(chuỗi trạng thái) tối ưu

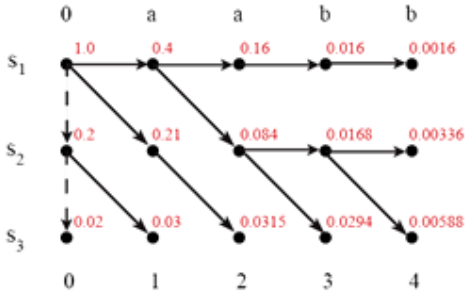
$$q_T^* = \psi_{t+1}(q_{t+1}^*), t = T-1, t-2, ..., 1 \quad (3\text{-}24)$$

+ Số phép tính $\approx N^2 T$

+ Ví dụ thuật toán Viterbi:



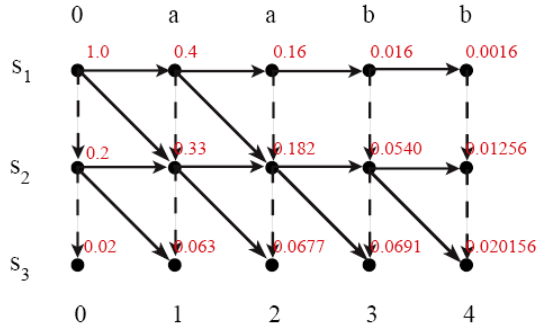
	0	*a*	*aa*	*aab*	*aabb*
s_1	1.0	s_1, a .4	s_1, a .16	s_1, b .016	s_1, b .0016
s_2	$s_1, 0$ .2	$s_1, 0$ .08 s_1, a .21 s_2, a .04	$s_1, 0$ .032 s_1, a .084 s_2, a .042	$s_1, 0$ .0032 s_1, b .0144 s_2, b .0168	$s_1, 0$ .00032 s_1, b .00144 s_2, b .00336
s_3	$s_2, 0$ .02	$s_2, 0$ .021 s_2, a .03	$s_2, 0$ .0084 s_2, a .0315	$s_2, 0$ .00168 s_2, b .0294	$s_2, 0$ .000336 s_2, b .00588



Hình 3- 6 Thuật toán Viterbi

+ Ví dụ so khớp sử dụng thuật toán tiến-lùi:

	0	*a*	*aa*	*aab*	*aabb*
s_1	1.0	s_1, a .4	s_1, a .16	s_1, b .016	s_1, b .0016
s_2	$s_1, 0$ .2	$s_1, 0$ .08 s_1, a .21 s_2, a .04	$s_1, 0$ .032 s_1, a .084 s_2, a .066	$s_1, 0$ .0032 s_1, b .0144 s_2, b .0364	$s_1, 0$ .00032 s_1, b .00144 s_2, b .0108
s_3	$s_2, 0$ .02	$s_2, 0$ .033 s_2, a .03	$s_2, 0$ .0182 s_2, a .0495	$s_2, 0$ .0054 s_2, b .0637	$s_2, 0$ .001256 s_2, b .0189



Hình 3- 7 so khớp sử dụng thuật toán tiến lùi

3.3.2 Ước lượng lại với thuật toán Baum-Welch

Ước lượng lại với thuật toán Baum-Welch sử dụng EM để xác định tham số ML

Xét toán tử $\xi_t(i)$ là xác suất của hệ thống ở trạng thái i tại thời điểm t và trạng thái j tại thời điểm t+1 với điều kiện có chuỗi quan sát O và mô hình Markov ẩn λ.

$$\xi_t(i, j) = P(q_t = s_i, q_{t+1} = s_j \mid O, \lambda) \quad (3\text{-}25)$$

Khi đó

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(i)}{P(O \mid \lambda)}$$

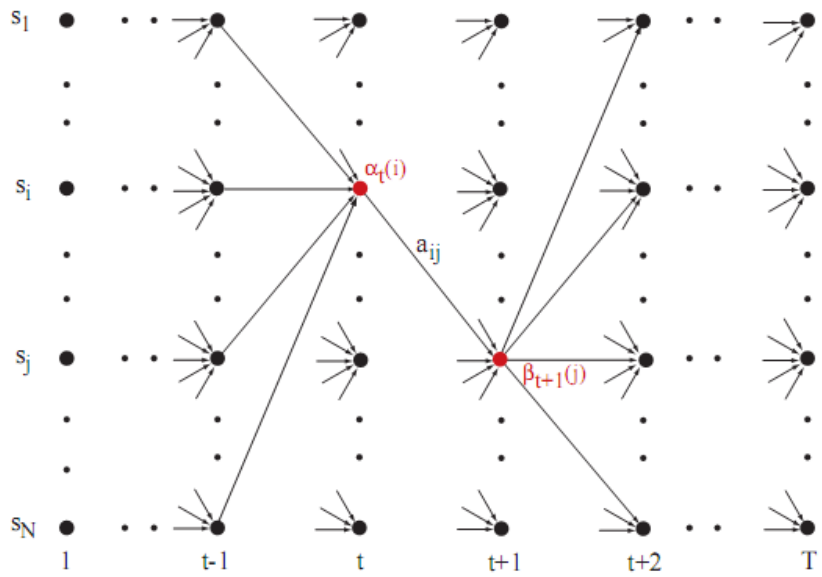
$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \quad (3\text{-}26)$$

Kết hợp $\gamma_t(i)$ và $\xi_t(i, j)$ chúng ta được:

$\sum_{t=1}^{T-1} \gamma_t(i)$ = số chuyển tiếp từ trạng thái si.

$\sum_{t=1}^{T-1} \xi_t(i,j)$ = số chuyển tiếp từ trạng thái si tới sj.

Thủ tục ước lượng lại Baum-Welch



Các biểu thức ước lượng lại với thuật toán Baum-Welch

$$\bar{\pi} = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_{ij} = \frac{\sum_{t=1, o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \quad (3\text{-}27)$$

Nếu $\lambda(A,B,\pi)$ là mô hình gốc và $\bar{\lambda}(\bar{A},\bar{B},\bar{\pi})$ là mô hình ước lượng lại, khi đó ta có thể chứng minh:

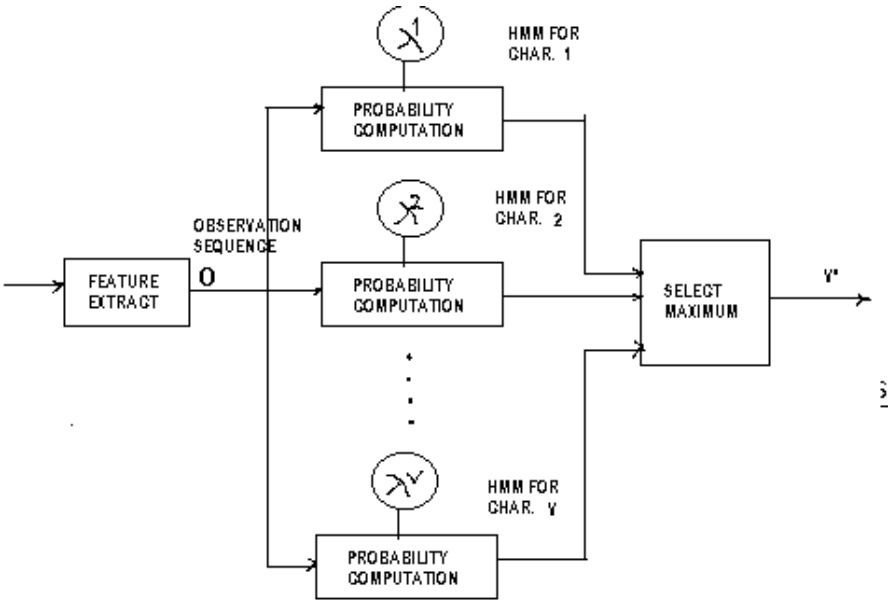
Mô hình gốc λ xác định điểm tới hạn của hàm có khả năng xảy ra, trong trường hợp $\bar{\lambda} = \lambda$. Hoặc:

Mô hình $\bar{\lambda}$ thích hợp hơn λ trong điều kiện $P(O \mid \bar{\lambda}) > P(O \mid \lambda)$

Chúng ta có thể tăng xác suất chuỗi quan sát O mà đã quan sát được từ mô hình nếu sử dụng lặp lại $\bar{\lambda}$ trong không gian λ và lặp lại việc ước lượng lại cho đến khi một số điểm tới hạn đạt được. Mô hình kết quả thu được gọi là mô hình Markov ẩn có khả năng xảy ra lớn nhất.

3.5 Ứng dụng của HMM trong nhận dạng chữ viết tay

Gần đây, phương pháp HMMs đã được sử dụng chủ yếu trong lĩnh vực nhận dạng chữ viết bao gồm cả chữ ký xác minh, nhận dạng ký tự, từ công nhận và như vậy. Trong phần này, ứng dụng của HMMs về vấn đề nhận dạng ký tự và các vấn đề công nhận từ chủ yếu là thảo luận.Sơ đồ khối của một điển hình HMM dựa trên hệ thống nhận dạng ký tự được thể hiện trong hình vẽ dưới. Trong hệ thống này cho mỗi ký tự HMM được xây dựng. Ví dụ, trong tiếng Việt, 66 HMMs được sử dụng để mô hình 33 ký tự của cả hai trường hợp chữ hoa và chữ thường. Quá trình nhận dạng là khá đơn giản. Các tính năng ký tự dạng chữ viết tay được mô hình hóa dưới dạng các dãy số được gọi là chuỗi quan sát O. Xác suất P (O | λ i) còn được hiểu là xác xuất tạo ra chuỗi quan sát O tương ứng với mỗi mô hình λ i. Các ký tự được nhận dạng sẽ là ký tự tương ứng với mô hình có xác xuất lớn nhất V = argmax [P (O | λ i). Hệ thống nhận dạng được mô hình hóa theo hình vẽ dưới đây [2]:



Hình 3- 8 Mô hình ứng dụng HMM trong nhận dạng chữ viết tay

Mỗi ký tự đầu vào của hệ thống được được mô hình hóa dựa trên các đặc điểm của từng ký tự đó thu được một chuỗi quan sát ký tự O, chuỗi quan sát này được so sánh với đặc điểm nhận dạng của từng chữ cái (đã được mô hình hóa bởi thuật toán Markov hay còn gọi là các HMM) để tìm ra ch1ữ cái có đặc điểm giống nhất (xác xuất giống cao nhất) để nhận diện ký tự quan sát được.

3.6 Giới hạn của các mô hình Markov ẩn

Trong bài báo "Maximum Entropy Markov Model for Information Extraction and Segmentation" Adrew McCallum đã đưa ra hai vấn đề mà các mô hình HMM truyền thống nói riêng và các mô hình sinh (generative models) nói chung gặp phải khi

gán nhãn cho dữ liệu dạng chuỗi.

Thứ nhất, để có thể tính được xác suất P(S, O) thông thường ta phải liệt kê hết các trường hợp có thể của chuỗi S và chuỗi O. Nếu như các chuỗi S có thể liệt kê được vì số lượng các trạng thái là có hạn thì trong một số ứng dụng ta không thể nào liệt kê hết được các chuỗi O vì dữ liệu quan sát là hết sức phong phú và đa dạng.

Để giải quyết vấn đề này, HMM phải đưa ra giả thiết về sự độc lập giữa các dữ liệu quan sát, đó là dữ liệu quan sát được tại thời điểm t chỉ phụ thuộc trạng thái tại thời điểm đó. Tuy vậy, với các bài toán gán nhãn cho dữ liệu dạng chuỗi, ta nên đưa ra các phương thức biểu diễn các dữ liệu quan sát mềm dẻo hơn như là biểu diễn dữ liệu quan sát dưới dạng các thuộc tính (features) không phụ thuộc lẫn nhau. Ví dụ với bài toán phân loại các câu hỏi và câu trả lời trong một danh sách FAQ, các thuộc tính có thể là bản thân các từ hay độ dài của dòng, số lượng các kí tự trắng, dòng hiện tại có viết lùi đầu dòng hay không, số các kí tự không nằm trong bảng chữ cái, các thuộc tính về các chức năng ngữ pháp của chúng… Rõ ràng những thuộc tính này không nhất thiết phải độc lập với nhau.

Vấn đề thứ hai mà các mô hình sinh gặp phải khi áp dụng vào các bài toánphân lớp dữ liệu dạng chuỗi đó là chúng sử dụng xác suất đồng thời để mô hình hóa,các bài toán có tính điều kiện.Với các bài toán này sẽ thích hợp hơn nếu ta dùng một mô hình điều kiện có thể tính toán P (S|O) trực tiếp thay vì P (S, O).

3.7 Mô hình Markov cực đại hóa Entropy (MEMM)

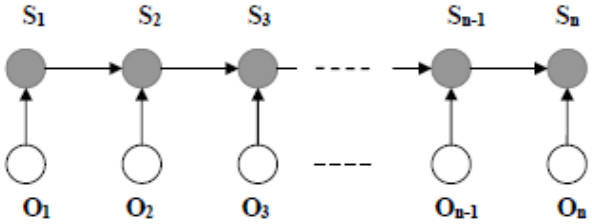
McCallum đã đưa ra một mô hình Markov mới - mô hình MEMM (Maximum Entropy Markov Model) như đáp án cho những vấn đề của mô hình Markov truyền thống.

3.7.1Tổng quan về mô hình Markov cực đại hóa Entropy (MEMM)

Mô hình MEMM thay thế các xác suất chuyển trạng thái và xác suất sinh quan sát trong HMM bởi một hàm xác suất duy nhất P (Si|Si-1, Oi) - xác suất để trạng thái hiện tại là Si với điều kiện trạng thái trước đó là Si-1 và dữ liệu quan sát hiện tại là Oi.

Mô hình MEMM quan niệm rằng các quan sát đã được cho trước và chúng ta không cần quan tâm đến xác suất sinh ra chúng, điều duy nhất cần quan tâm là các xác suất chuyển trạng thái. So sánh với HMM, ở đây quan sát hiện tại không chỉ phụ thuộc vào trạng thái hiện tại mà còn có thể phụ thuộc vào trạng thái trước đó, điều đó có

nghĩa là quan sát hiện tại được gắn liền với quá trình chuyển trạng thái thay vì gắn liền với các trạng thái riêng lẻ như trong mô hình HMM truyền thống.



Hình 3- 4 Đồ thị có hướng mô tả một mô hình MEMM

Áp dụng tính chất Markov thứ nhất, xác suất P(S|O) có thể tính theo công thức:

$$P(S \mid O) = P(S_1 \mid O_1) * \prod_{t=1}^{n} P(S_t \mid S_{t-1}, O_1) \quad (3\text{-}20)$$

MEMM coi các dữ liệu quan sát là các điều kiện cho trước thay vì coi chúng như các thành phần được sinh ra bởi mô hình như trong HMM vì thế xác suất chuyển trạng thái có thể phụ thuộc vào các thuộc tính đa dạng của chuỗi dữ liệu quan sát. Các thuộc tính này không bị giới hạn bởi giả thiết về tính độc lập như trong HMM và giữ vai trò quan trọng trong việc xác định trạng thái kế tiếp. Kí hiệu PSi-1(Si|Oi)=P(Si|Si-1,Oi). Áp dụng phương pháp cực đại hóa Entropy, McCallum xác định phân phối cho xác suất chuyển trạng thái có dạng hàm mũ như sau:

$$P_{S_{i-1}}(S_i \mid O_i) = \frac{1}{Z(O_i, S_{i-1})} \exp\left(\sum_a \lambda_a f_a(O_i, S_i)\right) \quad (3\text{-}21)$$

Ở đây, a, λ là các tham số cần được huấn luyện (ước lượng); Z (Oi, Si) là thừa số chẩn hóa để tổng xác suất chuyển từ trạng thái Si-1 sang tất cả các trạng thái Si kề đều bằng 1; fa (Oi, Si) là hàm thuộc tính tại vị trí thứ i trong chuỗi dữ liệu quan sát

và trong chuỗi trạng thái. Mỗi hàm thuộc tính fa (Oi,Si) nhận hai tham số, một là dữ liệu quan sát hiện tại Oi và một là trạng thái hiện tại Si. McCallum định nghĩa a=<b, Si>, ở đây b là thuộc tính nhị phân chỉ phụ thuộc vào dữ liệu quan sát hiện tại và Si là trạng thái hiện tại. Sau đây là một ví dụ về một thuộc tính b: Hàm thuộc tính fa (Oi, Si) xác định nếu b (Oi) xác định và trạng thái hiện tại nhận một giá trị cụ thể nào đó:

$$b(O_i) = \begin{cases} \text{1 nếu dữ liệu quan sát hiện tại là "the"} \\ \text{0 nếu ngược lại} \end{cases} \quad (3\text{-}22)$$

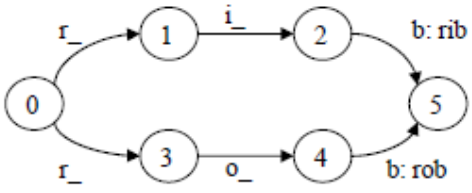
nếu ngược lại

$$f_a(O_i, S_i) = \begin{cases} \text{1 nếu } b(O_i) = 1 \text{ và } S_i = S_{i-1} \\ \text{0 nếu ngược lại} \end{cases} \quad (3\text{-}23)$$

Để gán nhãn cho dữ liệu, MEMM xác định chuỗi trạng thái S làm cực đại P(S|O) trong công thức (2.3).Việc xác định chuỗi S cũng được thực hiện bằng cách áp dụng thuật toán Viterbi như trong HMM.

3.7.2Vấn đề "label bias"

Trong một số trường hợp đặc biệt, các mô hình MEMM và các mô hình địnhnghĩa một phân phối xác suất cho mỗi trạng thái có thể gặp phải vấn đề "label bias". Ta hãy xem xét một kịch bản chuyển trạng thái đơn giản sau:



Hình 3- 5 Vấn đề "label bias"

Giả sử ta cần xác định chuỗi trạng thái khi xuất hiện chuỗi quan sát là "rob". Ở đây, chuỗi trạng thái đúng S là '0345' và ta mong đợi xác suất P (0345|rob) sẽ lớn hơn xác suất P(0125|rob).Áp dụng công thức (2.3), ta có:

$$P(0125|rob) = P(0) * P(1|0, r) * P(2|1, o) * P(5|2, b) \quad (3\text{-}24)$$

Vì tổng các xác suất chuyển từ một trạng thái sang các trạng thái kề với nó bằng 1 nên mặc dù trạng thái 1 chưa bao giờ thấy quan sát 'O' nhưng nó không có cách nào khác là chuyển sang trang thái 2, điều đó có nghĩa là P (2|1, x) =1 với x có thể là một quan sát bất kì. Một cách tổng quát, các trạng thái có phân phối chuyển với entropy thấp (ít đường đi ra) có xu hướng ít chú ý hơn đến quan sát hiện tại.

Lại có P (5|2, b) =1, từ đó suy ra: P (0125|rob) = P(0)*P(1|0,r). Tương tự ta cũng có P (0345|rob)=P (0)*P (3|0,r). Nếu trong tập huấn luyện, từ 'rib' xuất hiện thường xuyên hơn từ 'rob' thì xác suất P(3|0,r) sẽ nhỏ hơn xác suất P(1|0,r), điều đó dẫn đến xác suất P(0345|rob) nhỏ hơn xác suất P(0125|rob), tức là chuỗi trạng thái S=0125 sẽ luôn được chọn dù chuỗi quan sát là 'rib' hay 'rob'. Năm 1991, Léon Bottou đưa ra hai giải pháp cho vấn đề này.Giải pháp thứ nhất là gộp hai trạng thái 1, 3 và trì hoãn việc rẽ nhánh cho đến khi gặp một quan sát xác định (cụ thể ở đây là 'i' và 'o'). Đây chính là trường hợp đặc biệt của việc chuyển một automata đa định sang một automata đơn định. Nhưng vấn đề ở chỗ ngay cả khi có thể thực hiện việc chuyển đổi này thì cũng gặp phải sự bùng nổ tổ hợp các trạng thái của automata. Giải pháp thứ hai mà Bottou đưa ra là chúng ta sẽ bắt đầu mô hình với một đồ thị đầy đủ của các trạng thái và để cho thủ tục huấn luyện tự quyết định một cấu trúc thích hợp cho mô hình.Tiếc rằng giải pháp này sẽ làm mất tính đi tính có thứ tự của mô hình, một tính chất rất có ích cho các bài tóan trích chọn thông tin. Một giải pháp đúng đắn hơn cho vấn đề này là xem xét toàn bộ chuỗi trạng thái như một tổng thể và cho phép một số các bước chuyển trong chuỗi trạng thái này đóng vai trò quyết định với việc chọn chuỗi trạng thái. Điều này có nghĩa là xác suất của toàn bộ chuỗi trạng thái sẽ không phải được bảo tồn trong quá trình chuyển trạng thái mà có thể bị thay đổi tại một bước chuyển tùy thuộc vào quan sát tại đó.Trong ví dụ trên,

xác suất chuyển tại 1 và 3 có thể có nhiều ảnh hưởng đối với việc ta sẽ chọn chuỗi trạng thái nào hơn xác suất chuyển trạng thái tại 0.

3.8 Tóm tắt thông tin

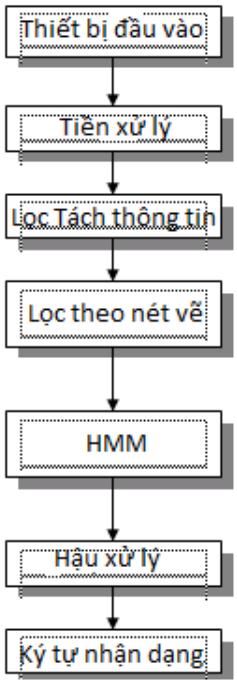
Trong chương này chúng ta đã tìm hiểu sâu về giải thuật Markov, giới thiệu về mô hình markov và ba bài toán đặt ra đối với mô hình này để chúng ta có sự nhìn nhận sâu sắc về mô hình và các ứng dụng của nó trong lĩnh vực nhận dang. Trong chương này chúng ta cũng mô tả tổng quan chức năng nhận dạng chữ viết dựa trên mô hình Markov ẩn. Chức năng này chúng ta sẽ tìm hiểu sâu hơn trong chương 4.

CHƯƠNG 4: ỨNG DỤNG HMM VÀO HỆ THỐNG NHẬN DẠNG

HMM ban đầu được nghiên cứu cho lĩnh vực nhận dạng tiếng nói, nhưng kỹ thuật này đã nhanh chóng được phát triển và ứng dụng trong các bài toán nhận dạng, trong đó có nhận dạng chữ viết và chiếm vị trí quan trọng trong lĩnh vực này. Chương này mô tả cách thức thực hiện việc nhận dạng chữ viết online dựa trên phương pháp tiếp cận HMMs. hệ thống nhận dạng được phát triển để nhận dạng các chữ viết in hoa tiếng Việt, độ chính xác của việc nhận dạng cũng được đề cập đến trong chương này.

4.1 Sơ đồ khối của hệ thống

Các sơ đồ khối của hệ thống nhận dạng được thể hiện trong hình 4-1. Việc nhận dạng có thể được mô tả như sau. Đầu tiên, thông tin bằng văn bản của từ này bị bắt bởi các thiết bị đầu vào và tiền xử lý để lấy ra các đặc tính của ký tự đầu vào. Tiền xử lý cũng loại bỏ thông tin không liên quan của các tín hiệu đầu vào. Thông tin lấy ra được sau bước này là chuỗi các ký tự mô hình hóa cho ký tự quan sát được và số nét vẽ tương ứng với ký tự đó nhằm tăng tính chính xác và tăng hiệu năng của hệ thống nhận dạng. Hai tính năng này là cơ sở để nhận dạng chữ viết dựa trên số nét vẽ ta có thể phân loại ký tự đầu vào sẽ nằm trong tập các ký tự có số nét vẽ đó việc này sẽ giảm thiểu việc so sánh ở bước tiếp theo.Bước kế tiếp theo sẽ thực hiện so sánh chuỗi ký tự đầu vào với tập các ký tự đã được mô hình hóa bằng HMM để tìm ra ký tự có xác xuất giống ký tự đầu vào nhất. Bước cuối cùng hệ thống xử lý để đưa ký tự nhận dạng đó hiển thị trên giao diện người sử dụng.Các chi tiết của từng khối chức năng trong hệ thống được thảo luận trong các phần tiếp theo.

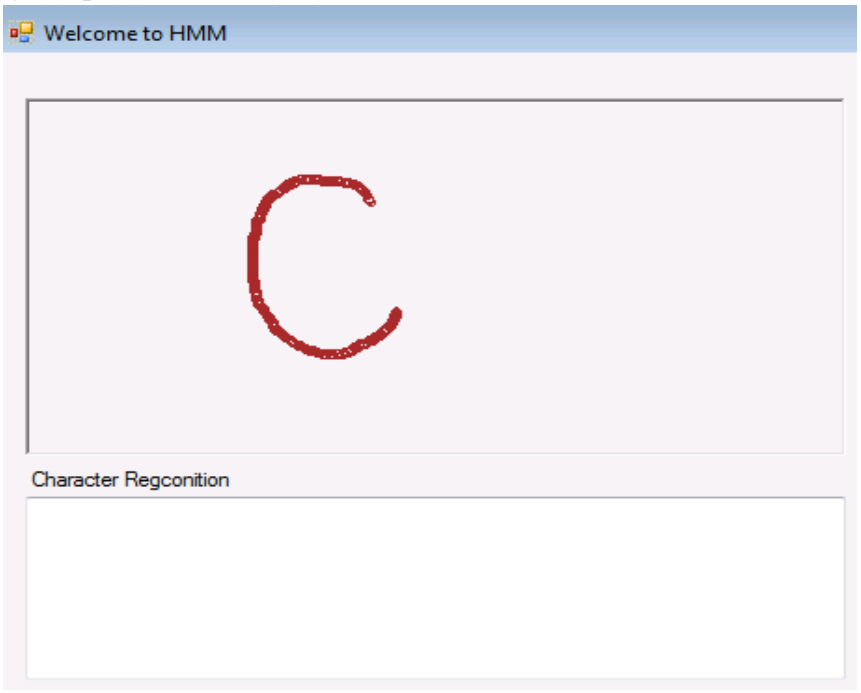


Hình 4- 1 Sơ đồ khối hệ thống nhận dạng chữ viết

4.2Thu thập dữ liệu đầu vào và xử lý dữ liệu

Thiết bị đầu vào trong luận văn này tác giả được tác giả thiết kế là một đối tượng picture box của Visual Studio, và lấy chuột máy tính để mô phỏng bút viết hình 4-2. Khi di chuột trên đối tượng Picture box cũng giống như việc viết tay trên giấy hay các thiết bị cảm ứng (tablet hoặc Smart phone hiện có trên thị trường). Mục đích là mô phỏng lại các chữ viết tay để làm yếu tố đầu vào cho hệ thống.Các dữ liệu thô, được lấy ra thông qua các phương thức của đối tượng picturebox bao gồm khi con chuột đặt xuống đối tượng, khi di chuột và khi nhâng chuột ra khỏi đối tượng, Đầu ra của bước này là tập hợp các tọa độ mà con chuột máy tính, số nét vẽ, trạng thái (con chuột tiếp xúc với đối tượng hay không) được mô tả theo tập hợp sau:

(X (t), y (t), p (t)) ∈ R2 x {0,1}, t = 1, 2... M.



Hình 4- 2 Nhập dữ liệu trên đối tượng PictureBox

trong đó (x (t), y (t)) là vị trí con chuột, trong khi p (t) là trạng thái con chuột tiếp xúc với đối tượng picturebox hay không ? là (p (t) = 1) tiếp xúc hoặc (p (t) = 0 không tiếp xúc) việc này giống như chiếc bút có viết lên tờ giấy hay được nhấc lên khỏi tờ giấy. Từ dữ liệu này nguyên các tính năng văn bản khác nhau có thể được thu được hoặc ngầm hay rõ ràng, bao gồm:

- Ký tự viết đã được mô tả bằng tập các điểm M (x (t), y (t)) nếu không tính đến trình tự của các điểm thì gọi là nhận dạng offline
- Quỹ đạo của con chuột trên đối tượng picturebox dựa trên các điểm M có tính trình tự của các điểm (x (1), y (1))... (x (M), y (M)).
- Số đặt con chuột (đặt bút) được biết đến, dựa trên các tham số p (t), p (t) thay đổi 1-0, Đây là căn cứ quan trọng để tính toán số lượng nét vẽ cho bước kế tiếp.
- Các thông tin về tốc độ viết có thể lấy ra dựa trên số lượng các điểm lấy mẫu bị bắt trên một đơn vị thời gian. Từ đó ta biết được người viết nhanh hay chậm.

Trong số các dữ liệu thu thập trên dữ liệu tốc độ viết không phải thông tin quan trọng đối với việc nhận dạng do đó sẽ được loại bỏ khi sau khi qua bộ tiền xử lý.

4.3 Tiền xử lý

Tiền xử lý là bước quan trọng trong hệ thống nhận dạng, tiền xử lý tốt sẽ tăng sự chính xác của hệ thống nhận dạng. Trong hệ thống, tiền xử lý bao gồm việc lấy ra tọa độ các điểm của ký tự đầu vào, loại bỏ dữ liệu xấu và phân đoạn như sau:

Hình 4- 3 Các bước tiền xử lý

4.3.1 Lấy mẫu các điểm

Mục đích của bước này là để giảm ảnh hưởng của tốc độ đối với việc nhận dạng. Khoảng cách giữa 2 điểm liên tiếp (x (t), y (t)) và (x (t +1), y (t +1)) được tính như sau:

$$d(t) = \sqrt{(x(t+1)-x(t))^2 + (y(t+1)-y(t))^2} \quad (4\text{-}1)$$

Nếu chọn d quá nhỏ số lượng điểm mẫu sẽ rất lớn khi đó khi thực hiện bước so sánh chuỗi ký tự đầu vào với các mẫu được mô hình hóa bằng HMM sẽ tốn hiệu năng của hệ thống. d khoảng cách do đó cần được chuẩn hóa và lựa chọn phù hợp để không làm quá tải hệ thống. Hình 4-4 cho thấy một ví dụ về mật độ các điểm lấy mẫu ký tự "C" dựa trên khoảng cách d với hai giá trị khác nhau.

Hình 4- 4 Ký tự "C" lấy mẫu với hai giá trị d khác nhau

P (t) là số điểm t của điểm N sau khi lấy mẫu bởi khoảng cách D, P (c) là các mẫu hiện tại bắt bởi con chuột, M là tổng số của mẫu chụp bởi máy tính bảng, khoảng cách điểm thuật toán bình thường có thể được tóm tắt như sau:

1. Initialization t =1; c=1; P(t)=P(c); c=c+1;

2. Repeat:

Calculate the distance d(t) between p(t) and p(c) using (4.1)

If d (t) < $d_{threshold}$ then

c=c+1;

else

{ t=t+1;

p(t)=p(c);

c=c+1;

N =t; }

Until c =M

Bước này đôi khi được gọi là bước lấy mẫu lại.

4.3.2 Loại bỏ các dữ liệu dư thừa

Mục đích của bước này là để loại bỏ các điểm dữ liệu xấu do quá trình lấy mẫu. Khi bắt đầu viết, người viết bi chuột trên đối tượng picturebox và kể từ khi lấy mẫu được thực hiện bất cứ khi nào bút là trong picturebox, trước khi cây bút hay con chuột xuống để bắt đầu viết một số mẫu đã được ghi nhận. Những mẫu này là các điểm dữ liệu xấu. Điều tương tự cũng xảy ra khi người viết bằng văn bản và di

chuyển bút vào trong không khí. Trước khi con chuột được di chuyển ra khỏi picturebox, một số mẫu cũng đã được thực hiện. Hình 4-5 là một minh hoạ của các điểm dữ liệu xấu khi các ký tự "C" được viết.

Hình 4- 5 minh họa các điểm dữ liệu xấu

Các điểm dữ liệu xấu được đại diện bởi một đường chấm chấm trong hình. Vì các điểm dữ liệu có các thông số đặt bút hay đặt chuột p (t) = 0 các thuật toán được sử dụng để loại bỏ chúng dễ dàng. Bằng việc phát hiện tất cả các dữ liệu đầu tiên và cuối cùng điểm của văn bản với p (t) = 0 và loại bỏ chúng. .

4.4 Lấy các thông tin từ ký tự quan sát được

Mục đích của của bước này là lấy ra các thông tin quan trọng của ký tự quan sát được để đưa vào bước tiếp theo. Hai việc quan trọng nhất trong khâu này là lấy ra được số nét viết và chuỗi mã hóa ký tự...

4.4.1 Lấy số nét vẽ

Toàn bộ các chữ viết tiếng việt được chia thành 5 nhóm dựa trên số lượng các nét vẽ của các ký tự:

• 1 nét: C, L, O, S, U, V, W, Z.

• 2 nét: B, D, G, J, K, M, N, P, Q, R, T, X, Y,Ô,Ơ.

• 3 nét: A, F, H, I,Đ.

• 4 nét: E,Â,Ă.

• 5 nét: Ê.

Như đã trình bày ở phần trước sau khi xử lý, mỗi ký tự sẽ được mã hóa dưới dạng một tập các điểm theo một trình tự có điểm bắt đầu và kết thúc, số nét vẽ của ký tự được xác dịnh theo giải thuật sau:

Với mỗi ký tự nhận dạng

Khởi tạo: số_nét_vẽ = 0;

on_mouse _down

{ số_nét_vẽ ++;

p(t)=1; // đánh dấu chuột đặt xuống đối tượng picturebox

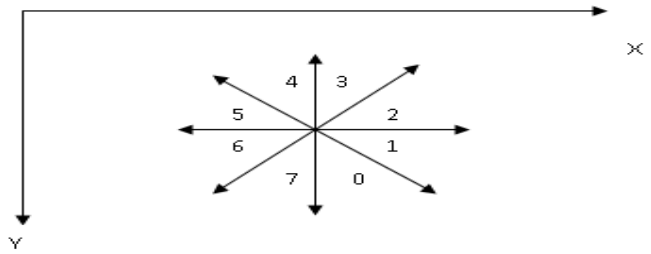
}

on_mouse_up: p(t)=0;

Việc sử dụng số nét vẽ làm giảm đáng kể khối lượng tính toán, ta sẽ đề cập đến việc này trong phần kế tiếp.

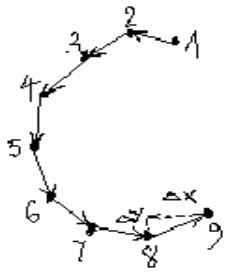
4.4.2 Mã hóa ký tự quan sát được

Tám hướng được sử dụng để xây dựng mã quỹ đạo khi di chuột trên đối tượng pictureBox để viết chữ. Hình 4.6 cho thấy 8 hướng trong X, Y tọa độ của màn hình máy tính.



Hình 4- 6 Tám hướng của chuỗi mã

Để xem cách này mã hóa các công trình, chúng ta hãy lấy ký tự "C" là một ví dụ. Giả sử rằng các ký tự "C" được đại diện bởi 9 điểm bằng văn bản chống chiều kim đồng hồ như thể hiện trong hình 4-7.



Hình 4- 7 Mã ký tự C

Quỹ đạo con chuột của các ký tự "C" được xác định bởi chuỗi các vectơ nối các điểm tiếp P (t) và P (t +1). Vector Mỗi xác định bởi một cặp (Δ x, Δ y): Δ x = x (t +1) x (t) và Δ y = y (t +1)-y (t). Mỗi cặp (Δ x, Δ y) tương ứng với một trong 8 hướng phụ thuộc vào dấu của Δ x, Δ y và mối quan hệ của họ như thể hiện trong hình 4.8.

Δx	Δy	Relation	Code
+	+	$\|\Delta x\| \geq \|\Delta y\|$	1
+	-	$\|\Delta x\| \geq \|\Delta y\|$	2
+	-	$\|\Delta x\| < \|\Delta y\|$	3
-	-	$\|\Delta x\| \leq \|\Delta y\|$	4
-	-	$\|\Delta x\| > \|\Delta y\|$	5
-	+	$\|\Delta x\| \geq \|\Delta y\|$	6
-	+	$\|\Delta x\| < \|\Delta y\|$	7
+	+	$\|\Delta x\| < \|\Delta y\|$	0

Hình 4- 8 Bảng mã từ quan hệ (Δ x, Δ y)

Các ký tự "C" được mã hóa theo cách trên có thể được đại diện bởi chuỗi mã chuỗi 5-6-6-7-0-1-1-2 trình tự này được sử dụng như trình tự của các chuỗi quan sát được đối với mỗi HMM. Một số nhận xét khi sử dụng mã hóa để mô hình hóa cho các ký tự dạng chữ viết tay:

- Các ký tự được viết theo cách khác nhau là khác nhau.
- Có một số cặp các ký tự có mã chuỗi trình tự rất giống nhau bao gồm: OC, DP, PR, JT O khác từ C ở một số điểm cuối cùng, D và P có thể được đại diện

bởi chuỗi mã cùng một chuỗi, R khác với P ở một số điểm cuối và J cũng khác với T trong một số điểm cuối cùng.

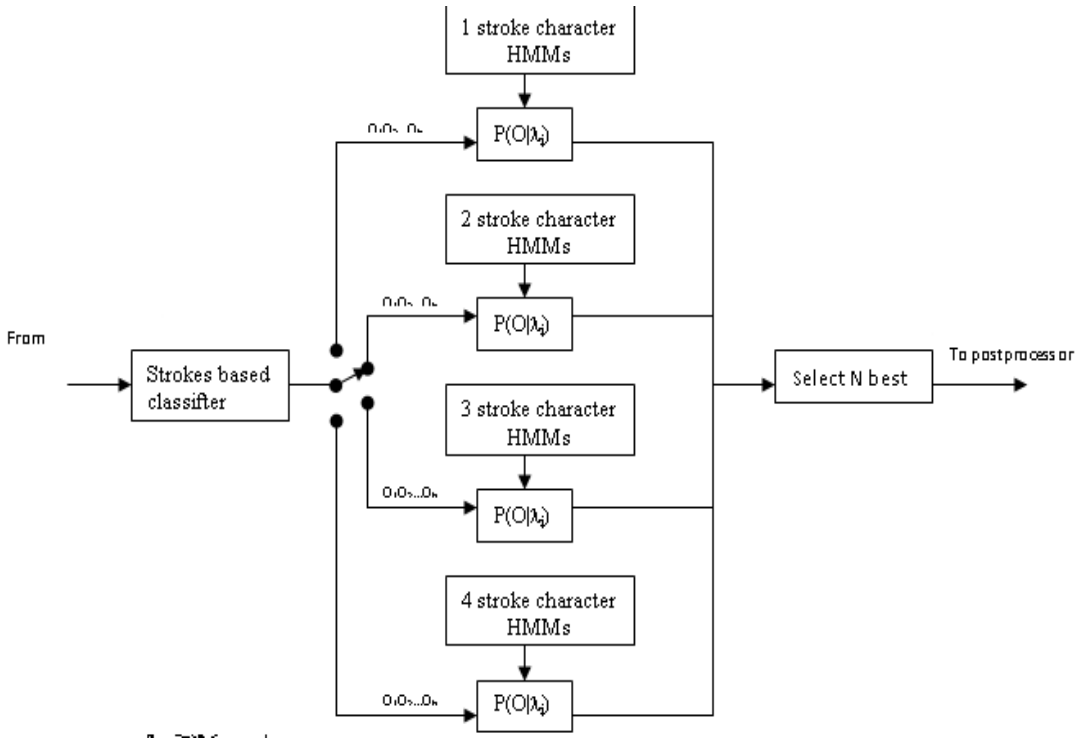
Tóm lại, tại đầu ra của tính năng giải nén các thông tin của mỗi ký tự trong chữ viết bao gồm số lượng của nét vẽ và mã chuỗi trình tự có sẵn cho quá trình tiếp theo.

4.5 Phân loại dữ liệu

Phân loại nhận dữ liệu từ các tính năng lấy các đặc tính trên, xử lý dữ liệu và tạo các dữ liệu đầu ra dưới hình thức xác suất của từng ký tự được nhận dạng. Quá trình phân loại bao gồm 2 mức: phân loại theo số nét vẽ và dùng HMMs để phân loại. Sơ đồ khối đầy đủ của phân loại cấp hai được hiển thị trong hình 4.9. Quá trình phân loại được mô tả ngắn gọn như sau:

Kết quả của bước trên ta thu được một chuỗi mã hóa ký tự quan sát được và số nét vẽ đây được coi là yếu tố đầu vào của quá trình phân loại, bước đầu tiên của quá trình phân loại là phân loại theo số nét vẽ ví dụ số nét vẽ là 4 khi đó hệ thống hiểu ký tự cần nhận dạng chỉ nằm trong ba ký tự E,Â,Ă mà không cần tính toán nhận dạng cho các ký tự còn lại, Bước thứ 2 hệ thống sẽ so sánh chuỗi mã hóa quan sát được với chuỗi mã hóa đã được mô hình hóa bởi HMM để tìm ra HMM có xác xuất

giống chuỗi quan sát được lớn nhất, từ đó quyết định ký tự nhận dạng là ký tự nào.



Hình 4- 9 Hai mức độ phân loại

Việc phân loại theo nét vẽ đã giảm đáng kể khối lượng tính toán của hệ thống, thay vì chuỗi mã quan sát được O ta đi đem so sánh với toàn bộ các HMM ứng với tất cả các chữ viết in hoa tiếng việt, ta chỉ so sánh với các HMM của các ký tự có số nét vẽ đó từ đó tìm ra xác xuất giống nhau giữa chuỗi O và các HMM. Bước này cũng làm tăng tính chính xác của hệ thống nhận dạng. PhầnTiếp theo, triển khai thực hiện HMMs bao gồm cả việc lựa chọn các thông số mô hình, train,… sẽ được thảo luận chi tiết.

4.5.1Lựa chọn mô hình và các thông số

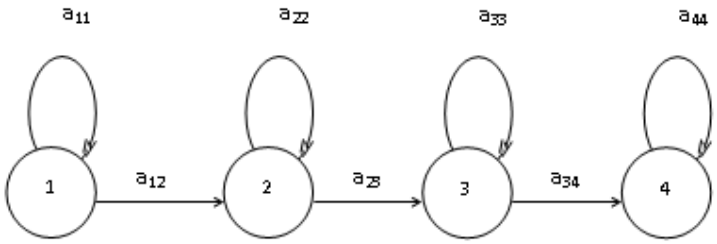
Trong hệ thống, mỗi ký tự được mô phỏng bằng cách rời rạc HMM $\lambda = (A, B, \pi)$ với các thuộc tính sau:

- Số lượng của các trạng thái N = 4, 8, 12 được thử nghiệm để xem ảnh hưởng của tham số này trên mô hình.

- Số quan sát các ký hiệu M = 8

  V = "0", "1", "2"... "6", "7"]

Các mô hình thử nghiệm mô hình là một mô hình dịch chuyển từ trái ->phải. Dưới sự kết hợp các trạng thái của chuỗi mã hóa với mô hình, các trạng thái có thể dịch chuyển sang trạng thái khác hoặc giữ nguyên (xem hình 4-10).



Hình 4- 10 Mô hình bốn trạng thái dịch chuyển

Trong chương 3, ta xem xét HMMs trong đó các quan sát đã liên kết với các trạng thái của mô hình. Trong hệ thống này, ta sử dụng một biến thể của HMMs trong đó quan sát được liên kết với các vòng cung của mô hình. Sự lựa chọn của nguyên mẫu mô hình này dẫn đến các thuộc tính sau đây của các ma trận A, B và π:

- aij = 0 j <i j> i +1, tức là, không có sự chuyển đổi trạng thái có chỉ số thấp hơn so với trạng thái hiện tại hoặc không cho phép nhẩy nhiều hơn 1 trạng thái. Đối với trạng thái cuối cùng ANN = 1. Hình thức của ma trận chuyển trạng thái cho các mô hình của con số 4,11 như sau:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\pi_i = \begin{cases} 0 & i \neq 1 \\ 1 & i = 1 \end{cases}$ trạng thái ban đầu bắt buộc phải là 1 (và kết thúc ở trạng thái N).

Phân bổ các ký tự của chuỗi quan sát được phân bố theo ij, B={bij(k)} trong đó bij(k) là xác xuất tạo ra ký tự quan sát được vk khi chuyển trạng thái từ i sang j

bij(k) = P[vk tại t|qt=SI và t+1|qt=Sj] $1 \le i, j \le N$; $1 \le k \le M$.

$$\sum_{k=1}^{M} b_{ij}(k) = 1$$

và

trong trường hợp N=4, M=8 ta có:

$$B = \begin{bmatrix} b_{11}(1) & b_{11}(2) & \dots & b_{11}(8) \\ b_{12}(1) & b_{11}(2) & \dots & b_{11}(8) \\ b_{22}(1) & b_{22}(2) & \dots & b_{22}(8) \\ b_{23}(1) & b_{23}(2) & \dots & b_{231}(8) \\ b_{33}(1) & b_{33}(2) & \dots & b_{33}(8) \\ b_{34}(1) & b_{34}(2) & \dots & b_{34}(8) \\ b_{44}(1) & b_{44}(2) & \dots & b_{44}(8) \end{bmatrix}$$

Để train một HMM cho mỗi ký tự, một tập các chuỗi ký tự mã của 40 ký tự được viết bởi một người được sử dụng. Mỗi ký tự được biểu diễn dưới dạng chuỗi mã có độ dài khác nhau. Độ dài các chuỗi từ mã trung bình là 50. Theo kinh nghiệm tập ký tự train khoảng 40 ký tự là tối ưu cho hệ thống [2]. Các chuỗi ký tự được sao lưu trong một file text khi bắt đầu train hệ thống sẽ đọc file này để lấy các chuỗi chuẩn này để thực hiện so sánh tìm ra xác xuất giống nhất với ký tự quan sát được đầu vào. Như vậy mỗi ký tự sẽ được nhận dạng bởi nhiều chuỗi mã hóa để tăng độ chính xác và tạo ra một HMM.

4.6 Mô Hình Training

Trước khi train các mô hình, các thông số của mô hình bao gồm cả các ma trận A và B đã được khởi tạo ngẫu nhiên. Sau đó, các thông số này được ước lượng để diễn tả các thuộc tính của tập trainiing sử dụng giải thuật Baum-Welch được

trình bày trong chương 3. Tuy nhiên bởi việc sử dụng các ký tự quan sát được kết hợp với các trạng thái trong mô hình, thuật toán này hay các thuật toán được trình bày trong chương 3 sẽ phải sửa đổi cho phù hợp. Trước hết, các thủ tục về phía trước được sửa đổi như sau:

Khởi tạo:

$$\alpha 1(t) = \pi i,\ 1 \le i \le N \qquad (4.2)$$

$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i)\, a_{ij} b_{ij}(O_t), \quad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix}$$

bằng phương pháp quy nạp ta có

(4. 1)

Kết thúc

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad (4.2)$$

Tương tự với việc sử đổi thuật toán backward:

Khởi tạo:

$$\beta T(i) = 1,\ 1 \le i \le N \qquad (4.3)$$

Quy nạp:

$$\beta t(i) = \sum_{j=1}^{N} a_{ij} b_{ij}(O_t) \beta_{t+1}(j),$$

$$t = T-1, T-2, \ldots, 1,\ 1 \le i \le N \qquad (4.4)$$

sử dụng xác suất forward αt (i) and backward $\beta t(i)$, $\gamma t(i)$ và $\xi t(i, j)$ được thay đổi như sau:

$$\gamma_t(i) = \frac{\alpha_t(i).a_{ii} b_{ii}(O_t) \beta_{t+1}(i)}{P(O \mid \lambda)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_{ij}(O_t) \beta_{t+1}(j)}{P(O \mid \lambda)} \qquad (4.5)$$

Cuối cùng các thông số được lượng hóa như sau:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$
$$\bar{b}_{ij}(k) = \frac{\sum_{t=1, O_t=k}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (4.6)$$

Quá trình train được kết thúc khi so sánh toàn bộ các ký tự trong chuỗi quan sát được. Đặt k là số ký tự trong chuỗi quan sát được ta mô tả chuỗi quan sát như sau:

$$O = [O(1), O(2),\ldots, O(k)] \quad (4.7)$$

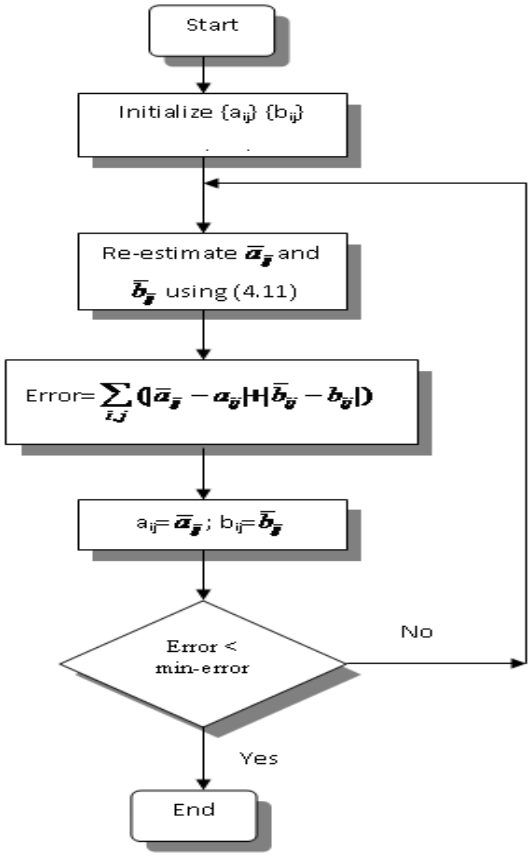
Mục đích của bước này là tìm ra tham số λ tối ưu nhất:

$$P(O|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda) = \prod_{k=1}^{K} P_k \quad (4.8)$$

Lượng hóa lại sử dụng công thức:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \xi_t^{\,k}(i,j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \gamma_t^{\,k}(i)}$$
$$\bar{b}_{ij}(l) = \frac{\sum_{k=1}^{K} \sum_{t=1, O_t=l}^{Tk-1} \xi_t^{\,k}(i,j)}{\sum_{k=1}^{K} \sum_{t=1}^{Tk-1} \gamma_t^{\,k}(i)} \quad (4.9)$$

Toàn bộ quá trình được mô tả trong hình 4-11.



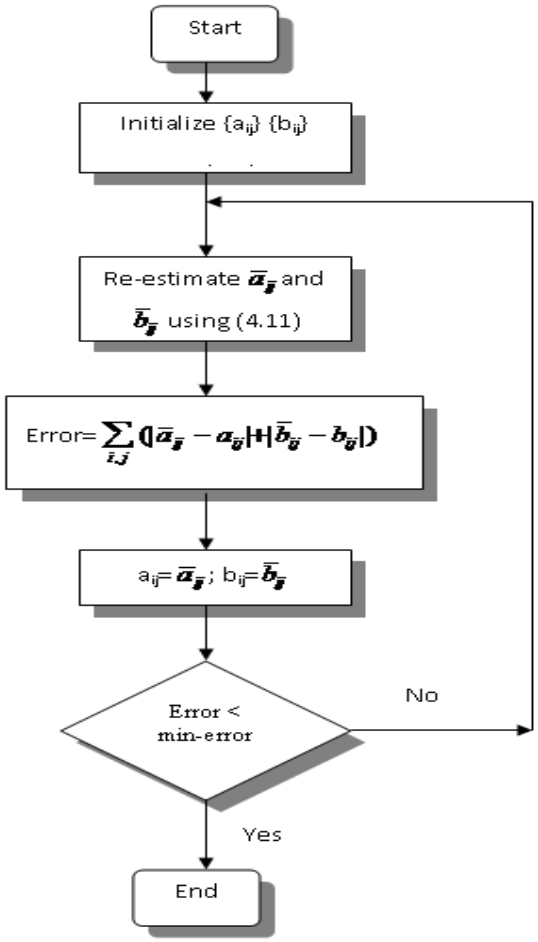
Hình 4- 11 Thuật Toán HMMS Training

Như thể hiện trong hình 4-11 quá trình lượng hóa giữ cho tỉ lệ sai số khi chuyển dịch ký tự trong chuỗi quan sát được với các HMM được giữ một sai số nhất định.. Sau khi HMM đã được trained, nó đã được sử dụng để tính toán xác suất của một chuỗi biểu tượng quan sát dựa trên các thông số của mô hình và forward thuật toán được sử dụng.

4.7. Postprocessor

Mục đích của bước này là để cải thiện tỉ lệ nhận dạng của hệ thống bằng cách sử dụng kiến thức về ngôn ngữ. Để nhận dạng ký tự dựa trên từ trong đó tỷ lệ nhận dạng hoàn toàn phụ thuộc vào việc nhận dạng các ký tự, nếu tỉ lệ nhận dạng của ký là 95% và chiều dài từ trung bình là 5, ví dụ như trong tiếng Anh, sau đó tỷ lệ từ

nhận được phiên dịch sang 95^5= 77 %.. Tại đầu vào của postprocessor, N ký tự ứng cử viên tốt nhất cho mỗi ký tự trong một từ ký tự L có sẵn. Ở đây, N = 3 được sử dụng. Từ những ứng cử viên, số lượng tối đa của từ đưa ra giả thuyết có thể là 3 * 3L. Ví dụ, nếu L = 5 ta có 729 từ giả thuyết trong đó một tối ưu từ có được lựa chọn và điều này là từ được nhận dạng. Từ tối ưu là từ có xác suất cao nhất trong số các từ được đưa ra giả thuyết có mặt trong từ điển. Để tìm một từ như vậy, chúng ta phải tìm kiếm trong không gian tìm kiếm các từ giả thuyết. Điều này không gian tìm kiếm sẽ phát triển theo cấp số nhân khi L là lớn và do đó nó sẽ mất một thời gian rất dài để tìm thấy một từ tối ưu. Vấn đề là làm thế nào để tìm kiếm lời tối ưu một cách hiệu quả. May mắn thay, có một kỹ thuật tìm kiếm tối ưu trong AI có thể được sử dụng cho mục đích này: chi nhánh và tìm kiếm ràng buộc. Thông thường, trong số các từ đưa ra giả thuyết là một từ tối ưu và chi nhánh và các công trình bị ràng buộc. Tuy nhiên, có những trường hợp trong đó không có từ trong những lời đưa ra giả thuyết là hiện tại trong từ điển. Do đó, không có giải pháp được tìm thấy trước khi tất cả các từ giả thuyết đã được thử. Trong những trường hợp này, postprocessor sẽ tìm thấy một từ trong từ điển là gần nhất từ giả thuyết rằng có xác suất cao nhất. Các sơ đồ khối của postprocessor được đưa ra trong hình 4-12. Trong luận văn này tác giả chỉ muốn đề cập đến việc nhận dạng ký tự, còn nhận dạng từ là hướng phát triển tiếp theo của luận văn này.



Hình 4- 12 Sơ đồ khối Postprocessor

4.8 Kết quả thực nghiệm

Để kiểm tra hiệu năng và tính chính xác của hệ thống với số lượng của các trạng thái được chọn N=8. Những thí nghiệm này được thực hiện duy nhất với một người viết. Bảng 4.13 cho thấy tỷ lệ nhận dạng của mỗi ký tự. Những kết quả này được tính toán dựa trên 10 chữ viết cho mỗi ký tự.

Character	N=8
A	100%
Â	100%

B	90%
C	85%
D	70%
Đ	90%
E	95%
Ê	95%
F	90%
G	100%
H	100%
I	95%
J	90%
K	100%
L	100%
M	100%
N	100%
O	90%
Ô	95%
P	80%
Q	90%

R	95%
S	95%
T	90%
U	95%
V	75%
W	100%
X	90%
Y	100%
Z	100%

Hình 4- 13 Bảng kết quả test với state=8

Từ những kết quả trong bảng 4.13, nó có thể được nhìn thấy rằng đối với N = 8, hệ thống có hiệu suất tốt nhất vì nó có thể nhận ra các ký tự khó hiểu như CO, DPR tốt hơn so với các hệ thống với N = 4, N = 12. Đối với các nhân vật khác, tất cả các hệ thống có thể nhận ra chúng với tỉ lệ nhận dạng rất cao. Hiệu quả của việc sử dụng một phân loại cấp độ đầu tiên là rõ ràng. Khi sử dụng một cấp độ đầu tiên phân loại C chỉ gây nhầm lẫn với O, nhưng mà không có nó C có thể được công nhận là sai G, O có thể được công nhận là Q và ngược lại, F có thể được công nhận là E, do đó tỉ lệ nhận dạng của O giảm từ 100 % (với phân loại cấp độ đầu tiên) đến 90% (không phân loại cấp độ đầu tiên), tỉ lệ nhận dạng của F giảm xuống từ 100% đến 80% và như vậy.

4.9 Tóm tắt thông tin

Trong chương này, việc thực hiện của HMM dựa trên dòng hệ thống nhận dạng văn bản đã được giới thiệu chi tiết. Hệ thống này bao gồm các khối chức năng tuần tự bao gồm cả máy tính bảng số hóa như một thiết bị đầu vào, một tiền xử lý, chiết tính năng, phân loại và postprocessor. Tiền xử lý thực hiển xử lý cả khoảng cách điểm bình thường, dữ liệu xấu loại bỏ và phân đoạn bằng cách sử dụng các phương pháp tiếp cận phân khúc bên ngoài. Các tính năng chiết xuất về số lượng các nét vẽ và tính năng mã chuỗi cho HMMs. Phân loại mức độ đầu tiên giúp cải thiện tỉ lệ nhận dạng, nhưng nó có thể làm cho hệ thống ít linh hoạt hơn. Cuối cùng, postprocessor, bằng cách sử dụng chi nhánh và ràng buộc tìm kiếm với từ điển tra cứu, đóng một vai trò rất quan trọng trong việc cải thiện tỉ lệ nhận dạng của toàn bộ hệ thống. Để đơn giản hóa các nhiệm vụ nhận dạng và có được một tỉ lệ nhận dạng cao.

KẾT LUẬN

Với sự phát triển mạnh mẽ của các thiết bị cảm ứng hiện nay, nhận dạng chữ viết online là một lĩnh vực rất hấp dẫn để nghiên cứu. Tuy nhiên ở Việt Nam hiện nay nhận dạng chữ viết chủ yếu được nghiên cứu theo hướng nhận dạng chữ viết in, hoặc nhận dạng chữ viết offline trên một văn bản hoặc ảnh đã viết sẵn, Trong khi nhận dạng chữ viết online có ưu điểm trong việc nhận định được trình tự viết, nhận dạng trực tiếp chữ viết người dùng viết,… nhưng lại gặp nhiều khó khăn về tốc độ xử lý dữ liệu, tốc độ viết. Luận án này tập trung nghiên cứu nhận dạng chữ viết online, giới thiệu về mô hình nhận dạng, tìm hiểu về mô hình Markov và ứng dụng mô hình Markov trong lĩnh vực nhận dạng chữ viết tay online. Từ các nghiên cứu trên, Ta có thể rút ra một số kết luận sau:

- Tiền xử lý và hậu xử lý không thể thiếu trong một hệ thống nhận dạng chữ viết bước này giúp cải thiện hiệu suất và nâng cao tính chính xác của toàn bộ hệ thống.
- Trích lọc chữ viết tay đóng một vai trò rất quan trọng trong việc công nhận tính chính xác. Bằng cách lấy ra các thuộc tính quỹ đạo viết và mô hình hóa thành các chuỗi mã kết hợp với số nét viêt cải thiện đáng kể tính chính xác của hệ thống, giúp hệ thống nhận dạng được nhiều phong cách viết.
- Rõ ràng là phương pháp tiếp cận HMMs là một phương pháp mạnh mẽ để nhận dạng chữ viết tay online. Bằng cách sử dụng sự dịch chuyển từ trái sang phải của 8 trạng thái HMMs rời rạc tương ứng với mô hình mỗi ký tự ta đã có thể đạt được độ chính xác cao khi nhận dạng chữ viết

Hướng phát triển tiếp theo của luận văn

Với những nội dung được trình bày trong đồ án, Chúng ta thấy đươc ý nghĩa của việc nhận dạng chữ viết online tuy nhiên trên thực tế nhận dạng chữ viết sẽ hoàn hảo hơn và có nhiều ứng dụng hơn khi chúng ta kết hợp việc nhận dạng chữ viết thành nhận dạng từ. Đây là hướng phát triển tiếp theo của luận văn. Cùng với hương

phát triển đó tác giả cũng tiếp tục nghiên cứu nhằm nâng cao tỷ lệ nhận dạng. Từ những kết quả thử nghiệm, nó có thể được nhìn thấy rằng tỉ lệ nhận dạng ký tự C, O, D, P thấp do sự giống nhau của các ký tự trong cách viết. Tỉ lệ nhận dạng của những ký tự này có thể được cải thiện bằng cách sử dụng các tính năng liên quan đến điểm bắt đầu và điểm kết thúc của văn bản. Các hệ thống trong luận án này cũng chưa linh hoạt do hạn chế về số lượng các phong cách viết cần phát triển hệ thống thích ứng được với nhiều phong cách viết. Mở rộng cho phép người viết viết các ký tự chữ thường. Nhiều mô hình như vậy, cần phải được sử dụng để mô hình 33 ký tự chữ thường. Sau đó hệ thống sẽ có thể nhận ra những từ bao gồm cả ký tự chữ hoa và chữ thường. Giải quyết được việc một người thuận tay trái hay tay phải khi viết. Tất cả những vấn đề trên sẽ được tác giả nghiên cứu và tiếp tục phát triển.

Qua đây tác giả cũng gửi lời cảm ơn chân thành đến TS.Phạm Ngọc Nam đã giúp đỡ tôi hoàn thành luận văn này. Tôi cũng xin chân thành cảm ơn anh César Roberto de Souza trường São Carlos (Brazil) đã cho phép tôi sử dụng lại một số hàm thư viện tính toán xác xuất trong luận văn này.

TÀI LIỆU THAM KHẢO

1. TS.Phạm Ngọc Nam, *Online handwritting regconition,*Thessis text, 2000.
2. Lawrence R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceeding of the IEEE, Vol.77, No.2, Feb. 1989.
3. Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. *Learning Hidden Markov Model Structure for Information Extraction*. AAAI 99 Workshop on Machine Learning for Information Extraction, 1999.
4. Http://vi.wikipedia.org/wiki/Mô_hình_markov_ẩn truy cập cuối cùng ngày 16/03/2012.
5. http://crsouza.blogspot.com/2010/03/hidden-markov-sequence-classifiers-in-c.html truy cập cuối cùng ngày 20/02/2012.
6. Michael A, Arbib (Ed.). *The Handbook of Brain Theory and Neural Networks,* Prentice-Hall, 1995.
7. Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.
8. Editors Jacob Benesty, M. Mohan Sondhi and Yiteng Huang, *Handbook of Speech Processing*, Springer-Verlag Berlin, 2008.
9. J.J. Hull. *Incorporation of markov model of language syntax in a text recognition algorithm.* In Proc. Symp. Document Analysis and Information retrieval, 1992.
10. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1999
11. Lawrence, Jeanette *Introduction to Neural Networks,* California Scientific Software Press. 1994.

PHỤ LỤC 1: CODE C# THỰC HIỆN HỆ THỐNG

```
// Hàm trích xuất dữ liệu và add vào thành một tập các điểm

  public void AddSample(Point p, int pressure)
      {
        int mindis = 10;
        double dis;
        pointlist[NoSamples] = p;
        //Pressure[NoSamples] = pressure;
        NoSamples++;
        //point distance normalization
        if (NoSamples > 2)
        {
                    dis = Math.Sqrt(GetYDerivative(NoSamples - 2) *
                  GetYDerivative(NoSamples - 2) + GetXDerivative(NoSamples - 2)
                  * GetXDerivative(NoSamples - 2));
          if (dis < mindis)
             NoSamples--;
        }
      }

// Hàm tính khoảng cách giữa hai điểm lân cận DeltaX, DeltaY
       public long GetYDerivative(int sample)
      {
        if (sample < (NoSamples - 1))
        {
          return (pointlist[sample + 1].Y - pointlist[sample].Y);
        }
        else
        {
```

```
        return (-1);
    }
}
public long GetXDerivative(int sample)
{
    if (sample < (NoSamples - 1))
    {
        return (pointlist[sample + 1].X - pointlist[sample].X);
    }
    else
    {
        return (-1);
    }
}
//hai hàm lấy tọa độ X,Y từ pointlist
public long GetX(int sample)
{
    if (sample < NoSamples)
    {
        return (pointlist[sample].X);
    }
    else
    {
        return (-1);
    }
}

public long GetY(int sample)
{
```

```
        if (sample < NoSamples)
        {
            return (pointlist[sample].Y);
        }
        else
        {
            return (-1);
        }
    }
// hàm tạo từ mã mô tả quỹ đạo viết
    public string createChan(int ns)
    {
        string s = "";
        for (int i = 0; i < ns; i++)
        {
            if (GetXDerivative(i) > 0.0)
            {
                if (GetYDerivative(i) > 0.0)
                {
                    if (GetXDerivative(i) > GetYDerivative(i))
                    {
                        s = s + "1-";
                    }
                    else
                    {
                        s = s + "0-";
                    }
                }
                else
```

```
    {
        if (GetXDerivative(i) > -GetYDerivative(i))
        {
            s = s + "2-";
        }
        else
        {
            s = s + "3-";
        }
    }
}
else
{
    if (GetYDerivative(i) > 0.0)
    {
        if (-GetXDerivative(i) > GetYDerivative(i))
        {
            s = s + "6-";
        }
        else
        {
            s = s + "7-";
        }
    }
    else
    {
        if (-GetXDerivative(i) > -GetYDerivative(i))
        {
            s = s + "5-";
```

```
                }
                else
                {
                    s = s + "4-";
                }
            }
        }

    }
    return s;
}
// Hàm tách chuỗi chancode thành ma trận
    private int[] decode(String sequence)
    {
        string[] elements = sequence.Split('-');
        int[] integers = new int[elements.Length];

        for (int j = 0; j < elements.Length-1; j++)
            integers[j] = int.Parse(elements[j]);

        return integers;
    }
// hàm add thêm điểm
public void AddSample(Point p, int pressure)
    {
        int mindis = 10;
        double dis;
```

```
            pointlist[NoSamples] = p;
            //Pressure[NoSamples] = pressure;
            NoSamples++;
            //point distance normalization
            if (NoSamples > 2)
            {
                dis = Math.Sqrt(GetYDerivative(NoSamples - 2) *
GetYDerivative(NoSamples - 2) + GetXDerivative(NoSamples - 2) *
GetXDerivative(NoSamples - 2));
                if (dis < mindis)
                    NoSamples--;
            }
        }

private void pictureBox1_MouseMove(object sender, MouseEventArgs e)

        {
            if (e.Button == MouseButtons.Left)
            {
                timer1.Enabled = false;
                timer1.Interval = WritingTimeOut;
                timer1.Enabled = true;

                Graphics g = pictureBox1.CreateGraphics();
                g.DrawEllipse(new Pen(Color.Brown), e.X, e.Y, 5, 5);
                Point p = new Point(e.X, e.Y);
                AddSample(p, 1);
            }
        }
        private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
```

```
        {
            if (NoSamples == 0)
            {
                Point p = new Point(e.X, e.Y);
                AddSample(p, 1);
            }
        }
        private StringBuilder OCR = new StringBuilder();
        private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
        {
            timer1.Interval = WritingTimeOut;
            timer1.Enabled = true;
            string chan = createChan(NoSamples);
            strokes++;
            Point p = new Point(e.X, e.Y);
            AddSample(p, 1);
        }

    public void train()
        {
            int rows = R.Count;
            int[][] sequences = new int[rows][];
            int[] labels = new int[rows];
            for (int i = 0; i < rows; i++)
            {
                HWOCRCharRecognition r = (HWOCRCharRecognition)R[i];
                string label = r.RecognizeAs;
                for (int j = 0; j < hmmc.Models.Length; j++)
                {
                    if (hmmc.Models[j].Tag.Equals(label))
```

```
            {
                labels[i] = j;
                break;
            }
        }
        sequences[i] = decode(r.chancode);
    }
    int iterations = 100;
    double limit = 0;
    for (int i = 0; i < 10; i++)
    {
        hmmc.Learn(sequences, labels, iterations, limit);
    }
}
private void btnCreate_Click(object sender, EventArgs e)
{
    if (PCRFPath == null)
    {
        MessageBox.Show("Please input HMM file");
        return;
    }
    LoadCharRecognitionTable();
    ArrayList A = getclassbystroke(strokes, R);
    string c = "";
    for (int i = 0; i < A.Count; i++)
    {
        HWOCRCharRecognition r = (HWOCRCharRecognition)A[i];
        c += r.RecognizeAs;
    }
```

```
int a = A.Count;
List<string> s = new List<string>();
for (int i = 0; i < a; i++)
{
   HWOCRCharRecognition r = (HWOCRCharRecognition)A[i];
   s.Add(r.RecognizeAs);
}
List<string> sd = s.Distinct().ToList();
string[] categories = new string[sd.Count];
int[] states = new int[sd.Count];
for (int i = 0; i < sd.Count; i++)
{
   categories[i] = sd[i];
   states[i] = 2;
}
string chuoi = "";
for (int i = 0; i < categories.Length; i++)
{
   chuoi += categories[i];
}
//MessageBox.Show(chuoi);
hmmc = new MarkovSequenceClassifier(sd.Count, 8, states, categories);
train();
double likelihood;
string sequencec = createChan(NoSamples);
MessageBox.Show("Chuoi ma hoa: "+sequencec);
int label = hmmc.Compute(decode(sequencec), out likelihood);
string assignedLabel = hmmc.Models[label].Tag as string;
regco = assignedLabel + "  ";
```

```
            txtCR.Text =regco;
            clean();
        }

        // Viet Ham doc tu file text
        private void LoadCharRecognitionTable()
        {
            R.Clear();
           FileStream fs = new FileStream(PCRFPath, FileMode.Open,
FileAccess.Read);
            StreamReader sr = new StreamReader(fs,Encoding.Default);
            while (sr.Peek() > 0)
            {
                string Temp = sr.ReadLine();
                string[] Temps = Temp.Split('#');
                HWOCRCharRecognition CR = new
HWOCRCharRecognition(Temps[0].ToString(), Int32.Parse(Temps[1]),
Temps[2].ToString());
                R.Add(CR);
            }
            sr.Close();
            fs.Close();
        }

       public ArrayList getclassbystroke(int strokes,ArrayList RC)
        {
            ArrayList A = new ArrayList();
            for (int i = 0; i < RC.Count; i++)
            {
                HWOCRCharRecognition r = (HWOCRCharRecognition)R[i];
```

```
            if (r.strokes == strokes)
            {
                A.Add(r);
            }
        }
        return A;
    }

// Hàm save chuỗi chancode

private void button42_Click(object sender, EventArgs e)
    {
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            PCRFPath = openFileDialog1.FileName;
            FileStream fs = new FileStream(PCRFPath, FileMode.Append);
            StreamWriter sw = new StreamWriter(fs,Encoding.Default);
            string s = chancode_save + "#" + strokes_save.ToString() + "#" +
textBox1.Text;
            sw.WriteLine(s);
            sw.Flush();
            sw.Close();
            fs.Close();
         }
        MessageBox.Show("input to text file successed");
        this.Close();
    }
```