第7章　おわりに

単語音声認識の実験はいかがでしたか？

私がこの実験を計画するにあたり，いろいろな選択肢がありました．フリーソフトを組み合わせて入力した音声を文字にする「音声タイプライタ」を作る，音声認識ソフトウェアを利用してコンピュータを操作したりロボットを動かす，などは受講者にとって魅力的な実習内容だと思います．実際に行うことも可能です．しかし，2 コマ× 3 回の時間でこのような実習を行おうとすると，”既存ソフトウェアをうまく使って組み合わせて応用した”的な内容になってしまいます．

大学では，そのようなことよりも**音声認識の基本原理そのものを学ぶ**べきだと思いました．本実験では，音声認識の原理そのもの，すなわちブラックボックスをホワイトボックスとして理解できるような実習を行うことを狙いとしました．その目的のため，ソースコードは個々のプログラムあたりせいぜい数百行に収めること，全て開示すること，自分でコンパイルできることとしました．

メディア情報学実験のテーマとして音声認識実験をこの制約で組み立てるとなると，音声タイプライタは実施不可能でした．音声タイプライタを実装するためには，単語 HMM を用意してもダメで，音素 HMM を学習し，3 万語程度以上の単語辞書を作成し，単語同士のつながりを規定するための文法規則あるいは統計情報を作成し，これらの情報を用いることによって入力音声から得られる数百の認識候補から答えを効率的に選びだすためのアルゴリズムが必要です．このような機能を備えたソフトウェアを作るとすると，専門の研究者による長年の開発が必要で，ソースコードの量は膨大になります．例えば，フリーの音声認識ソフトウェアである HTK（Hidden Markov Model ToolKit，`https://htk.eng.cam.ac.uk/`）

や Julius（https://julius.osdn.jp/）のソースコードはコメントも含めて C 言語で約 11〜12 万行もあります．音声認識に必要なことはほぼ全部できる汎用のツールなので，その機能を実現するために多量の記述が必要なのです．さらに，音素 HMM の学習データを作成するためには，そこそこの長さがある数百種類の文を録音し，音素の種類と位置を示した情報を付与する必要があります．手慣れたエキスパートでも，この作業だけでかなりの日数を要します．

スペクトル分析，HMM などの原理（数理）は教科書的に説明し，音声認識の実行は HTK や Julius などのフリーソフトを利用するような実習のイメージがちらっと頭をよぎったことがあります．でも，それでは原理と実装の間の溝は埋まりません．

あれこれ考えてみて，結局，研究室（髙木研究室とその母体である電通大尾関研究室（1989 年〜2008 年））で受け継がれている単語認識のプログラム群を利用することになりました．研究用に書かれたプログラムなので，最小限の入出力処理以外は，教科書に書かれた原理（数理）を実行する数値計算のソースコードしかなく，ソースコードのファイルを単独でコンパイルすれば動く，小さくて綺麗なプログラムが揃っています．そこで，このプログラム群からいくつかピックアップし主として入出力処理を改良したプログラム，および IED での音声入力用に新たに書いたプログラムを合わせて，単語認識実験を行うことにしたのです．

HMM 算法に関する実習プログラムは上坂先生尾関先生の『パターン認識と学習のアルゴリズム』[9] を参考に作成しました．この実習のソースコードは音声入力，音声区間検出など含めたプログラムも全部合わせ，コメントも含めて約 6700 行です．最も大きいソースコードは単語 HMM の学習プログラム `train.c` の 1529 行です．

受講生が「学んだ音声認識アルゴリズムを用いて実際に単語認識を行うソースコードを理解」できるようにする目標をもって教材を作りましたが，残念ながら現状では中途半端になってしまいました．第 3 週の単語認識の実習に関わる内容です．単語 HMM の学習を行うプログラム `train` に用いている Baum-Welch アルゴリズムでは連続 HMM 用の計算式（5.4.2 節）を用いています．本当は，式 5.29〜式 5.43 までの計算式を理解してもらって，穴埋め式の実習を行ってもらいたかったのですが，実習の時間が足りないため割愛しています．また，単語認識を行うビタビアルゴリズムのプログラム（`~/asr/wrecog/program/vtb.c`）はテキストにその原理を解説（5.3.3 節）してますが，実習の課題からは除外してあります．これを含めて多少の事項は割愛しても，**認識アルゴリズムを C 言語のソースコードとして把握可能な規模に収め，自分の音声を用いて音声認識を行う実習を優先**させたためです．ビタビアルゴリズムも単語音声を認識するために多次元次数ベクトルの扱いを行うと，時間内に消化しきれなくなってしまいます．そこで，HMM の学習とそれを用いた認識の基本原理は，演習的設定の条件で実習を行ってもらい，実際の単語 HMM の学習とそれを用いた単語認識は，出力確率の記述は複雑になるけれどもアルゴリズムは同じ，という説明にとどめています．実はコメントを含めてわずか 484 行のプログラム（`~/asr/wrecog/program/_recog.c`）なので，ちょっと頑張れば，どこでどの計算を行っているのかくらいのことは，

分かってもらえるものと思っています．

On-The-Fly 単語認識では，音声区間検出の閾値の設定に苦労したことと思います．本実験で用いた検出法が最も簡単なものだったからかもしれません．音声区間検出は音声認識の性能を左右する極めて重要な技術です．認識したい音声をその他の信号から区別するのは、実はとても難しいのです．古くから研究が進められていて，現在も新しい方法が提案されています．最近，音声区間検出手法を評価するためのデータとツールが作られました [21]．この古くて新しい技術に興味がある人は，まずは文献 [25] を読んでみると良いでしょう．

音声認識の研究に興味を持ったのであれば，まず「荒木雅弘，イラストで学ぶ音声認識」（文献 [28]）を手にとってみるとよいでしょう．比較的最近の音声認識技術の全般が分かりやすく解説されています．いま世間で音声認識というと，話した言葉が文字に変換されるスマートフォンに搭載されているソフトウェアをイメージするでしょう．フリーのソフトウェアを利用すれば，そのような音声認識を自分のパソコンでも実行できます．興味がある人には荒木雅弘『フリーソフトでつくる音声認識システム（第 2 版）〜パターン認識・機械学習の初歩から対話システムまで』[29] がお薦めです．

音声認識の基礎から最新の手法までを学びたい方は高島遼一『Python で学ぶ音声認識（機械学習実践シリーズ)』[34] が良いでしょう．近年、画像像処理，パターン認識，機械学習に広く使われている Python を用いて，音声分析の基本，DP マッチング，GMM-HMM，DNN-HMM，End-to-End モデルによる連続音声認識までを実例を用いて実行し、その仕組みを体験することができます．GitHub にソースコードが載っています．

中川 聖一編著『音声言語処理と自然言語処理 (増補)』[31] は音声言語（音声を媒介とする意図伝達手段）と文字言語（文字を媒介とする意図伝達手段）の工学的応用（音声認識、音声合成、機械翻訳、検索など）を目的とした基礎技術について，有機的に関連させて解説している良書です．各章に章末問題とその解答が巻末に設けられている他，音声言語処理と自然言語処理に有用な各種フリーソフトウェアが紹介されています．

限られた実習時間で音声認識を理解して（すくなくとも理解したつもりになって）もらうために工夫をしているつもりですが，いかがだったでしょうか？ 率直なご意見をお待ちしています．

髙 木 一 幸　　2014 年 11 月 4 日（2024 年 10 月 09 日改訂）
takagi@uec.ac.jp　www.takagi.inf.uec.ac.jp