UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

# A Critical Analysis of Design Flaws in the Death Star

Luke Skywalker
99652154

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Dr O. W. Kenobi

October 2099

# Acknowledgements

I would like to thank my dog, Muffin. I also would like to thank the inventor of the incubator; without him/her, I would not be here. Finally, I would like to thank Dr Herman Kamper for this amazing report template.

UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

# Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   *Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.*

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   *I agree that plagiarism is a punishable offence because it constitutes theft.*

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   *I also understand that direct translations are plagiarism.*

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   *Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism*

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   *I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.*

| | |
|---|---|
| Studentenommer / *Student number* | Handtekening / *Signature* |
| Voorletters en van / *Initials and surname* | Datum / *Date* |

# Abstract

## English

The English abstract.

## Afrikaans

Die Afrikaanse uittreksel.

# Contents

# List of Figures

# List of Tables

# Nomenclature

### Variables and functions

$p(x)$ Probability density function with respect to variable $x$.

$P(A)$ Probability of event $A$ occurring.

$\varepsilon$ The Bayes error.

$\varepsilon_u$ The Bhattacharyya bound.

$B$ The Bhattacharyya distance.

$s$ An HMM state. A subscript is used to refer to a particular state, e.g. $s_i$ refers to the $i^{\text{th}}$ state of an HMM.

$\mathbf{S}$ A set of HMM states.

$\mathbf{F}$ A set of frames.

$\mathbf{o}_f$ Observation (feature) vector associated with frame $f$.

$\gamma_s(\mathbf{o}_f)$ A posteriori probability of the observation vector $\mathbf{o}_f$ being generated by HMM state $s$.

$\mu$ Statistical mean vector.

$\Sigma$ Statistical covariance matrix.

$L(\mathbf{S})$ Log likelihood of the set of HMM states $\mathbf{S}$ generating the training set observation vectors assigned to the states in that set.

$\mathcal{N}(\mathbf{x}|\mu, \Sigma)$ Multivariate Gaussian PDF with mean $\mu$ and covariance matrix $\Sigma$.

$a_{ij}$ The probability of a transition from HMM state $s_i$ to state $s_j$.

$N$ Total number of frames or number of tokens, depending on the context.

$D$ Number of deletion errors.

$I$ Number of insertion errors.

$S$ Number of substitution errors.

### Acronyms and abbreviations

| | |
|---|---|
| AE | Afrikaans English |
| AID | accent identification |
| ASR | automatic speech recognition |
| AST | African Speech Technology |
| CE | Cape Flats English |
| DCD | dialect-context-dependent |
| DNN | deep neural network |
| G2P | grapheme-to-phoneme |
| GMM | Gaussian mixture model |
| HMM | hidden Markov model |
| HTK | Hidden Markov Model Toolkit |
| IE | Indian South African English |
| IPA | International Phonetic Alphabet |
| LM | language model |
| LMS | language model scaling factor |
| MFCC | Mel-frequency cepstral coefficient |
| MLLR | maximum likelihood linear regression |
| OOV | out-of-vocabulary |
| PD | pronunciation dictionary |
| PDF | probability density function |
| SAE | South African English |
| SAMPA | Speech Assessment Methods Phonetic Alphabet |

# Chapter 1

# Introduction

The last few years have seen great advances in speech recognition. Much of this progress is due to the resurgence of neural networks; most speech systems now rely on deep neural networks (DNNs) with millions of parameters [?, 1]. However, as the complexity of these models has grown, so has their reliance on labelled training data. Currently, system development requires large corpora of transcribed speech audio data, texts for language modelling, and pronunciation dictionaries. Despite speech applications becoming available in more languages, it is hard to imagine that resource collection at the required scale would be possible for all 7000 languages spoken in the world today.

I really like apples.

## 1.1. Section heading

This is some section with two table in it: Table 1.1 and Table 1.2.

**Table 1.1:** Performance of the unconstrained segmental Bayesian model on TIDigits1 over iterations in which the reference set is refined.

| Metric | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| WER (%) | 35.4 | 23.5 | 21.5 | 21.2 | 22.9 |
| Average cluster purity (%) | 86.5 | 89.7 | 89.2 | 88.5 | 86.6 |
| Word boundary $F$-score (%) | 70.6 | 72.2 | 71.8 | 70.9 | 69.4 |
| Clusters covering 90% of data | 20 | 13 | 13 | 13 | 13 |

**Table 1.2:** A table with an example of using multiple columns.

| | Accuracy (%) | | |
|---|---|---|---|
| Model | Intermediate | Output | Bitrate |
| Baseline | 27.5 | 26.4 | 116 |
| VQ-VAE | 26.0 | 22.1 | 190 |
| CatVAE | 28.7 | 24.3 | 215 |

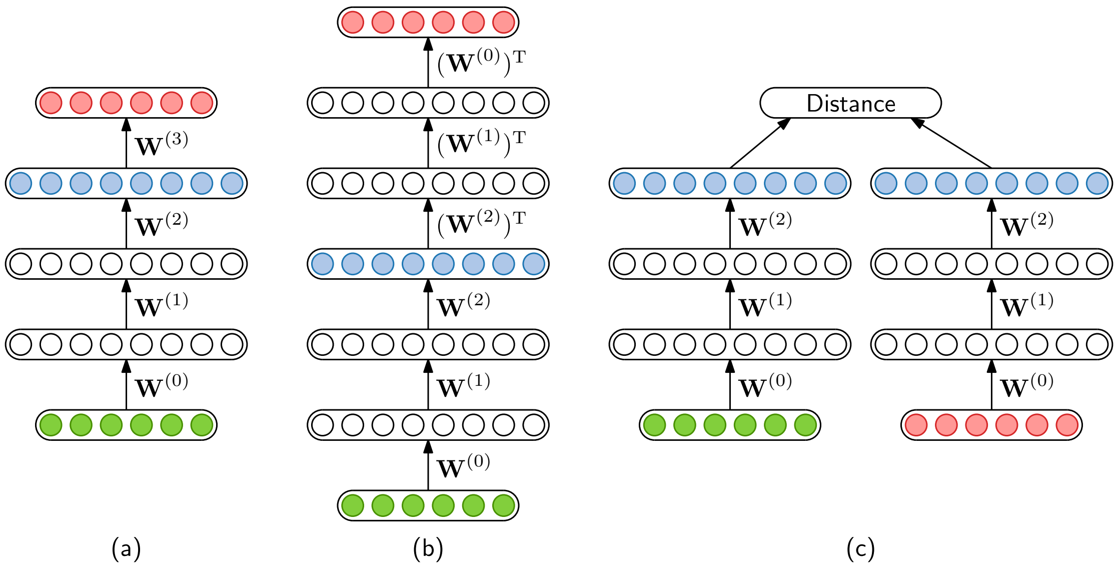

**Figure 1.1:** (a) The cAE as used in this chapter. The encoding layer (blue) is chosen based on performance on a development set. (b) The cAE with symmetrical tied weights. The encoding from the middle layer (blue) is always used. (c) The siamese DNN. The cosine distance between aligned frames (green and red) is either minimized or maximized depending on whether the frames belong to the same (discovered) word or not. A cAE can be seen as a type of DNN [**?**].

This is a new page, showing what the page headings looks like, and showing how to refer to a figure like Figure 1.1.

The following is an example of an equation:

$$P(\mathbf{z}|\boldsymbol{\alpha}) = \int_{\boldsymbol{\pi}} P(\mathbf{z}|\boldsymbol{\pi})\, p(\boldsymbol{\pi}|\boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\pi} = \int_{\boldsymbol{\pi}} \prod_{k=1}^{K} \pi_k^{N_k} \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}\, \mathrm{d}\boldsymbol{\pi} \tag{1.1}$$

which you can subsequently refer to as (1.1) or Equation 1.1. But make sure to consistently use the one or the other (and not mix the two ways of referring to equations).

# Chapter 2

# Body

## 2.1. Model Definition

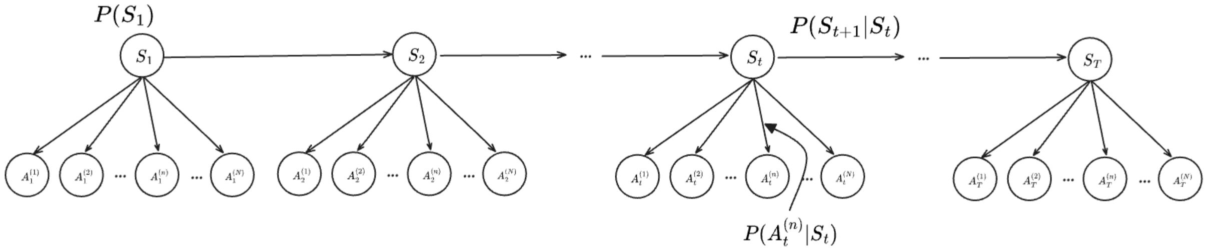

**Figure 2.1:** I am a caption below the figure of course

This is a reference to Figure 2.1
RVs:

- Hidden Drought State RVs $\equiv S_t = \{1, 2, \ldots, m\}$
- Attribute RVs $\equiv A_t^{(n)} = \{1, 2, \ldots, C_n\}$

Some further notation:

- $\mathbf{S}_{1:T} = \{S_1, S_2, \ldots, S_T\}$
- $A_{1:T} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_T\}$
  - Where $\mathbf{A}_t = \{A_t^{(1)}, A_t^{(2)}, \ldots, A_t^{(N)}\}$

### 2.1.1. Joint Distribution

$$
\begin{aligned}
&p(S_1, S_2, \ldots, S_T, A_1^{(1)}, A_1^{(2)}, \ldots, A_1^{(N)}, A_2^{(1)}, \ldots, A_T^{(N)}) \\
&= p(S_1, S_2, \ldots, S_T, \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_T) \\
&= p(\mathbf{S}_{1:T}, A_{1:T}) \\
&= p(S_1) \cdot \prod_{t=1}^{T-1} p(S_{t+1} \mid S_t) \cdot \prod_{n=1}^{N} \prod_{t=1}^{T} p(A_t^{(n)} \mid S_t)
\end{aligned}
$$

## 2.2. Factors

Priors

| $S_1$ | $p(S_1)$ |
|---|---|
| 1 | $\pi_1$ |
| 2 | $\pi_2$ |
| $\vdots$ | $\vdots$ |
| $m$ | $\pi_m$ |

Transition

| $S_t$ | $S_{t+1}$ | $p(S_{t+1} \mid S_t)$ |
|---|---|---|
| 1 | 1 | $a_{1,1}$ |
| 1 | 2 | $a_{1,2}$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| 1 | $m$ | $a_{1,m}$ |
| 2 | 1 | $a_{2,1}$ |
| 2 | 2 | $a_{2,2}$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $m$ | $m$ | $a_{m,m}$ |

$$\equiv \quad P^1 = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,m} \end{bmatrix}$$

Emission

| $A_t^{(n)}$ | $S_t$ | $p(A_t^{(n)} \mid S_t)$ |
|---|---|---|
| 1 | 1 | $b_1^{(n)}(1)$ |
| 1 | 2 | $b_2^{(n)}(1)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| 1 | $m$ | $b_m^{(n)}(1)$ |
| 2 | 1 | $b_1^{(n)}(2)$ |
| 2 | 2 | $b_2^{(n)}(2)$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $C_n$ | $m$ | $b_m^{(n)}(C_n)$ |

## 2.3. EM Theory

- $\mathcal{H} = (S_t)_{t=1}^T$
- $\mathcal{D} = (\mathbf{A}_t)_{t=1}^T$
- $\Theta = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$
- $\boldsymbol{\theta}_1 = \{\pi_1, \pi_2, \dots, \pi_m\} \equiv S_1$ Priors

- $\boldsymbol{\theta}_2 = \{a_{1,1}, a_{1,2}, \dots, a_{m,m}\} = P^1 \equiv$ Transition Probabilities
- $\boldsymbol{\theta}_3 = \{b_1^{(n)}(1), b_2^{(n)}(1), \dots, b_1^{(N)}(m)\} = P^1 \equiv$ Emission Probabilities

### 2.3.1. E-Step

Hold $\Theta$ fixed and choose $q$ such that

$$\begin{aligned} q(\mathcal{H}) &= p(\mathcal{H} \mid \mathcal{D}, \Theta) \\ &= p(\mathbf{S}_{1:T} \mid A_{1:T}, \Theta) \end{aligned}$$

### 2.3.2. M-Step

Hold $q$ fixed and optimise $\mathscr{L}(q, \Theta)$ w.r.t $\Theta$

After some math, this means finding $\Theta$ such that:

$$\begin{aligned} \Theta &= \underset{\Theta}{\operatorname{argmax}} Q(\Theta) \\ &= \underset{\Theta}{\operatorname{argmax}} \sum_{\mathcal{H}} q(\mathcal{H}) \cdot \log p(\mathcal{D}, \mathcal{H} \mid \Theta) \end{aligned}$$

## 2.4. Update Equations

Priors:

$$\pi_i^{\text{new}} = q(S_1 = i) \tag{2.1}$$

Transition Probabilities:

$$a_{ij}^{\text{new}} = \frac{\sum_{t=1}^{T-1} q(S_t = i, S_{t+1} = j)}{\sum_{t=1}^{T-1} q(S_t = i)} \tag{2.2}$$

Emission Probabilities:

$$b_i^{(n)}(j)^{\text{new}} = \frac{\sum_{t=1}^{T} q(S_t = i) \cdot I(A_t^{(n)} = j)}{\sum_{t=1}^{T} q(S_t = i)} \tag{2.3}$$

We can now reference these equations by their label: Equation 2.1, Equation 2.2 or Equation 2.3. This is wicked, lemme tell you

## 2.5. FIGURES TIME

things before figure

## 2.6. Determining $m$

We are using AIC, BIC and maximum log likelihood to do this. Here are the formulas:

$$AIC = -2 \cdot \log L(\Theta) + 2p$$
$$BIC = -2 \cdot \log L(\Theta) + p \cdot \log k$$

Where:

- $L(\Theta) \equiv$ the maximized value of the likelihood function for the estimated model
- $p \equiv$ the number of free parameters,
- $k \equiv$ the number of data points.

The idea is that we are going to sweep $m$, this means creating many models with different values of $m$, and choose the model that minimizes both AIC & BIC, whilst maximising $L(\Theta)$. Here is a more comprehensive criteria for choosing a particular $m$:

1. Primary: lowest BIC (preferred if you want parsimony, BIC penalizes complexity strongly).
2. Secondary: lowest AIC.
3. Also look at the log-likelihood curve: if log $L(\Theta)$ improves only marginally as $m$ increases, choose the simpler model (elbow rule).

But okay, lets see what the particular values are for our BIC & AIC calculations

### 2.6.1. What is $k$

We will first look at $k \equiv$ the number of data points: Not sure, can either see each observation as a vector, therefore we have $T$, or we can see each $a_t^{(n)}$ as an observation, and in that case we would have $T \times N$ observations.

### 2.6.2. What is $p$

- Next we look at $p \equiv$ the number of free parameters. Look more into this. See Occons Razor (A principled method to model selection and how BIC is an approximation of this), in the, use $m - 1$, so we will have:

$$\begin{aligned} p &= (m-1) + m(m-1) + \sum_{n=1}^{N} m(C_n - 1) \\ &= m^2 - 1 + m\sum_{n=1}^{N}(C_n - 1) \end{aligned}$$

### 2.6.3. How To Get $\ell(\Theta)$

This is the real kicker...

Of course the likelihood is the probability our data is observed, this means finding:

$$\begin{aligned} \ell(\Theta) &= p(A_1^{(1)} = a_1^{(1)}, A_1^{(2)} = a_1^{(2)}, \dots, A_T^{(N)} = a_T^{(N)} \mid \Theta) \\ &= p(A_{1:T}^{\text{obs}} \mid \Theta) \end{aligned}$$

Note: Observed Data $= (a_1^{(1)}, a_1^{(2)}, \dots, a_T^{(N)})$

To do this, we will use the Forward Algorithm (also note that have this (obs) superset makes things very verbose and will be omitted from here on out):

We begin by defining the factor $p(\mathbf{A}_t \mid S_t = i)$

- Recall, $\mathbf{A}_t = \{A_t^{(1)}, A_t^{(2)}, \dots A_t^{(N)}\}$
- Because of our model and the independent properties and things, we know that:

$$\begin{aligned} p(\mathbf{A}_t \mid S_t = i) &= p(A_t^{(1)}, A_t^{(2)}, \dots, A_t^{(N)} \mid S_t = i) \\ &= p(A_t^{(1)} \mid S_t = i) \cdot p(A_t^{(2)} \mid S_t = i) \dots p(A_t^{(N)} \mid S_t = i) \\ &= \prod_{n=1}^{N} p(A_t^{(n)} \mid S_t = i) \\ &= \prod_{n=1}^{N} e_i^{(n)}(a_t^{(n)}) \end{aligned}$$

Lets now look at the Forward Algorithm

- We first define

$$\begin{aligned} \alpha_t(i) &= p(S_t = i, A_{1:t}^{\text{obs}} \mid \Theta) \\ &= p(S_t = i, \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_t \mid \Theta) \end{aligned}$$

- We begin at $t = 1$:

$$\begin{aligned} \alpha_1(i) &= p(S_1 = i, A_{1:1}^{\text{obs}} \mid \Theta) \\ &= p(S_1 = i, \mathbf{A}_1 \mid \Theta) \\ &= p(S_1) \cdot p(\mathbf{A}_1 \mid S_1 = i, \Theta) \end{aligned}$$

- Then to move to the next time step:

$$\alpha_{t+1}(j) = (\sum_{i=1}^{m} \alpha_t(i) \cdot p_{i,j}) \cdot p(\mathbf{A}_{t+1} \mid S_{t+1} = j, \Theta)$$

- Thus, by the end we will have $\alpha_T(i) = p(A_{1:T}, S_T = i \mid \Theta)$
- Then finally we can obtain our likelihood by marginalising out $S_T$

$$\begin{aligned}\sum_{i=1}^{m} \alpha_T(i) &= \sum_{S_t} p(A_{1:t}, S_t \mid \Theta) \\ &= p(A_{1:t} \mid \Theta) \\ &= \ell(\Theta)\end{aligned}$$

Okay, lets regroup. Just so we fully get it. This is done after our model has been fitted and we already have our final $\Theta$. There is also no factors going on here. Our parameters ($\Theta$) are the probabilities in themselves. So when we observed our data $p(A_t^{(n)} = a_t^{(n)} | S_t = i)$ we are in essence choosing the value $b_i^{(n)}(a_t^{(n)})$ (Note, these $b$'s will have to change to $e$ or something...). And thus we have actual values we are playing with, thus pure math, not factors.

## 2.7. Meditating a little bit more on model output

## 2.8. What I have been doing

We are computing the pointwise marginal MAP, often called the Maximum Posterior Marginal (MPM) rule. For each time $t$, we pick:

$$\hat{s}_t = \underset{s}{\operatorname{argmax}}\; p(S_t = s \mid A_{1:T}, \Theta)$$

The MPM picks the most likely state at each time independently — which can lead to an impossible or very unlikely global sequence (eg. $S_t \equiv$ Very Wet, then $S_{t+1} \equiv$ Very Dry). It maximizes expected per-time classification accuracy, but it does not maximize the joint posterior probability of the entire sequence.

## 2.9. Viterbi Algorithm

The paper recommends the Viterbi algorithm, which finds

$$\mathbf{s}^* = \underset{\mathbf{S}_{1:T}}{\operatorname{argmax}}\; p(\mathbf{S}_{1:T} \mid A_{1:T}, \Theta)$$

ie. The single state sequence with the highest joint posterior probability. That sequence respects transitions and is temporally coherent.

Okay, Just a little reminder, we have used the LBU paired with EM. I spoke to my professor and he mentioned that because of the model structure, we are actually constructing a Junction Tree meaning we get exact inference. Additionally, because of how the junction tree we start from the leaf nodes, the formulation of using the JTREE vs Forward-Backward is actually the exact same (Check math behind this...).

Anyway, we have our model output now $p(\mathbf{S}_{1:T} \mid A_{1:T}, \Theta)$ which is an exact measure. How do i know get the output of my model. The paper I am implementing says this: ”With the estimated optimal DNBC parameters, the most probable path of the latent drought state that maximizes P(A— · ) together with the probability of each state at every time step can be obtained using the Viterbi algorithm (Rabiner 1989).” Right now I am simply taking the maximum confidence for each $p(S_t \mid A_{1:T}, \Theta)$. This is most probably wrong. What must I do, explain to me what i must do and why what im doing is wrong (if it is wrong.)

## 2.10. THIS NEEDS TO BE CLEANED UP, NOT SURE WHAT THESE THINGS BELOW ARE

Reminder: These are factors we have available:

- $q(\mathcal{H}) = p(\mathbf{S}_{1:T} \mid A^{\text{obs}}_{1:T}, \Theta)$
- $p(S_1)$
- $p(S_{t+1} \mid S_t)$
- $p(A_t^{(n)} \mid S_t)$

We want log $\ell(\Theta) = \log p(A^{\text{obs}}_{1:T} \mid \Theta)$

Math to get there:

1. Start With Each $p(A_t^{(n)} \mid S_t)$
2. For Each $A_t^{(n)}$
   (a) Observe data point $a_t^{(n)}$ to get: $p(A_t^{(n)} = a_t^{(n)} \mid S_t)$
   (b) Get $p(A_t^{(n)} = a_t^{(n)})$ understanding that:
   $$p(A_t^{(n)} = a_t^{(n)}) = \sum_{S_t} p(A_t^{(n)} = a_t^{(n)} \mid S_t) \cdot p(S_t)$$
   This $p(S_t)$ is our $q(\mathcal{H})$
3. Then of course, we multiply these to get
   $$p(A^{\text{obs}}_{1:T}) = \prod_{n=1}^{N} \prod_{t=1}^{T} p(A_t^{(n)} = a_t^{(n)})$$
4. These will likely underflow so we insert log now:
   $$\log \ell(\Theta) = \log p(A^{\text{obs}}_{1:T}) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log p(A_t^{(n)} = a_t^{(n)})$$

Thoughts?

If I can still use the forward equations, let me know. Otherwise I need to calculate the full joint distr and marginalise out?

A little bit embarrassing but how exactly do we get the log likelihood, the naive way. My understanding is that we:

1. Calculate the full joint distribution:

$$p(\mathbf{S}_{1:T}, A_{1:T}) = p(S_1) \cdot \prod_{t=1}^{T-1} p(S_{t+1} \mid S_t) \cdot \prod_{n=1}^{N} \prod_{t=1}^{T} p(A_t^{(n)} \mid S_t)$$

2. Marginalise out all $S_t$:

$$p(A_{1:T} \mid \Theta) = \sum_{S_{1:T}} p(\mathbf{S}_{1:T}, A_{1:T} \mid \Theta)$$

3. Then Observe Actual Data and sum the probs?

$$\text{Likelihood} = \sum p(A_{1:T} = \mathcal{D})??$$

I don't know what you mean by me being stuck. I get parameters which are the probabilities to the factors I am looking for. The model output is the max value $S_t$ which i get from my $q$ function. Let me know if I am overlooking something.

With regards to the AIC & BIC calcs. you have here $\ell(\Theta) = \sum_{S_{1:T}} p(\mathbf{S}_{1:T}, A_{1:T})$ but this leaves us with a factor not a single value? This is required for AIC and BIC which are single values? This is why I thought you must sum over observations?

Okay Ill stop faffing. Forward-Backward is new, I didnt want to waste a time sink learning it. Especially because this LBU + EM is much more flexible of a route which is good for me since I plan to expand this model. One idea I have is to introduce second-order markov property to the thing. Is Forward-Backward still feasible for a DNBC with the second order markov property? If not I am sticking to LBU. And thus maybe need an alternative to AIC and BIC. But let me know your thoughts.

## 2.11. Questions

1. Forward-Backward Equations. I have to right? From what I can see it applies to second order as well when we vectorise our states?
    - Its only really a problem to try and get $log\ \ell(\Theta)$ for AIC and BIC. Besides this it works fine?
    - is extracting $S_t$ from $q(\mathcal{H})$ fine and correct? Since we want $p(S_t)$ not $p(S_t \mid A_{1:T})$...
2. Based on this as about LBU things:
    (a) emdw has this `#include "lbu2_cg.hpp"`. What is this??
    (b) Ask about LTRIP vs other methods → Other Methods: BETHE, JTREE
3. Breaking symmetry for the priors $p(S_1)$. Is it necessary?
4. With regards to BIC & AIC, we need $k \equiv$ Number of Free Parameters (See calcs on `model-dev-clean` pg 11)
5. Discrete vs Cts Attribute RVs. See `model-dev-clean` pg 10
6. `main.cpp` line 951

## 2.12. Forward-Backward

- Initial: $\pi_i = P(S_1 = i)$.
- Transition: $a_{ij} = P(S_{t+1} = j \mid S_t = i)$.
- Emission Probabilities: $b_i^{(n)}(j) = p(A_t^{(n)} = j \mid S_t = i)$.

### 2.12.1. Define Forward Values

$$\alpha_t(i) = p(S_t = i, A_{1:t}^{\text{obs}} \mid \Theta)$$

### 2.12.2. Define Backward Values

$$\beta_t(i) = p(A_{t+1:T}^{\text{obs}} \mid S_t = i, \Theta)$$

### 2.12.3. Updates

First Define:

$$\gamma_t(i) = p(S_t = i \mid A_{1:T}^{\text{obs}}, \Theta) = \frac{p(S_t = i, A_{1:T}^{\text{obs}} \mid \Theta)}{p(A_{1:T}^{\text{obs}} \mid \theta)} = \frac{\alpha_t(i)\beta_t(i)}{\sum\limits_{k=1}^{m} \alpha_t(k)\beta_t(k)}$$

Node marginals:

$$\gamma_t(i) = P(S_t = i \mid A_{1:T}, \Theta) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{k=1}^{m} \alpha_t(k)\beta_t(k)}.$$

Pairwise marginals:

$$\xi_t(i,j) = P(S_t = i, S_{t+1} = j \mid A_{1:T}, \Theta) = \frac{\alpha_t(i)a_{ij}P(A_{t+1} \mid S_{t+1} = j)\beta_{t+1}(j)}{\sum\limits_{p=1}^{m}\sum\limits_{q=1}^{m} \alpha_t(p)a_{pq}P(A_{t+1} \mid S_{t+1} = q)\beta_{t+1}(q)}.$$

1. Forward $\alpha_t(i) = P(A_{1:t}, S_t = i \mid \theta)$ and backward $\beta_t(i) = P(A_{t+1:T} \mid S_t = i, \theta)$. Use scaling or log-space. 2. Responsibilities:

$$\gamma_t(i) \equiv P(S_t = i \mid A_{1:T}, \theta) = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_\ell \alpha_t(\ell)\beta_t(\ell)}.$$

3. Pairwise responsibilities:

$$\xi_t(i,j) \equiv P(S_t = i, S_{t+1} = j \mid A_{1:T}, \theta) = \frac{\alpha_t(i)\,a_{ij}\,P(A_{t+1} \mid S_{t+1} = j)\,\beta_{t+1}(j)}{\sum_{p,q} \alpha_t(p)\,a_{pq}\,P(A_{t+1} \mid S_{t+1} = q)\,\beta_{t+1}(q)}.$$

For multiple independent sequences $n = 1, \ldots, N$, compute $\gamma_t^{(n)}$ and $\xi_t^{(n)}$ per sequence and sum where needed.

"Baum's auxiliary function" = EM $Q$-function

$$Q(\theta \mid \theta^{\text{old}}) \;=\; \mathbb{E}_{S_{1:T} \mid A_{1:T}, \theta^{\text{old}}}[\log P(S_{1:T}, A_{1:T} \mid \theta)]\,.$$

Expanding the complete-data log-likelihood under the DNBC factorization gives three decoupled blocks (initial, transition, emission). Maximizing $Q$ with simplex constraints $\sum_i \pi_i = 1$, $\sum_j a_{ij} = 1$, $\sum_k b_{i,m}(k) = 1$ via Lagrange multipliers yields normalized expected counts. That's "Baum–Welch". Same thing you should get if you derive EM straight.

M-step (discrete case, no priors)

Single sequence (replace sums with $\sum_n$ for multiple sequences):

**Initial:**

$$\pi_i^{\text{new}} = \gamma_1(i).$$

**Transition:**

$$a_{ij}^{\text{new}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}.$$

**Emission for each attribute $m$ and value $k$:**

$$b_{i,m}^{\text{new}}(k) \;=\; \frac{\sum_{t=1}^{T} \gamma_t(i)\,\mathbf{1}\{A_t^{(m)} = k\}}{\sum_{t=1}^{T} \gamma_t(i)}.$$

# Chapter 3

# Summary and Conclusion

# Bibliography

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

# Appendix A

# Project Planning Schedule

This is an appendix.

# Appendix B

# Outcomes Compliance

This is another appendix.