1 Statistical Sequence Modelling

Mark Gales

Speech recognition and speech synthesis can be viewed as examples of machine learning classification and regression respectively. One of the reasons for speech processing being a distinct research area is the need to handle the sequential nature of both the observation sequence, $\boldsymbol{Y}_{1:T}$, and the word sequence $\boldsymbol{w}_{1:L}$. This report discusses speech processing within a machine learning framework, and approaches that allow the sequential nature of the data to be handled.

1.1 Language Models: A Sequence Model

This report is primarily concerned with sequence models for speech recognition and speech synthesis. which are both sequence to sequence models. As a way of illustrating some of the underlying models that are used, a single sequence task is first examined, language modelling.

In language modelling the aim is to map the L length sequence of words $\boldsymbol{w}_{1:L} = w_1, \ldots, w_L$ to a single value that represents the probability of the language model, with parameters [1] $\boldsymbol{\lambda}$, generating the word sequence, $P(\boldsymbol{w}_{1:L})$. For simplicity of notation unless the model parameters are needed to specify different models or configurations they are omitted from equations. The probability of the word sequences is often estimated using an N-gram approximation where the prediction of the next word is only dependent on the preceding $N-1$ words [2]

$$P(\boldsymbol{w}_{1:L}) = \prod_{i=1}^{L} P(w_i|\boldsymbol{w}_{1:i-1}) \approx \prod_{i=1}^{L} P(w_i|\boldsymbol{w}_{i-N:i-1}) \tag{1.1}$$

this form of approximation is essential to enable variable length word sequences to be modelled.

The simplest approach is to use the counts derived from the training corpus, with appropriate techniques adopted to ensure robustness of the estimates. As an alternative it is possible to use a neural network to predict the probabilities (Bengio, Ducharme, Vincent & Jauvin 2003). A more interesting form of language model is based on Recurrent Neural Networks (RNNs) (Mikolov, Karafiát, Burget, Cernockỳ & Khudanpur 2010).

A problem with the n-gram approximation is that it restricts the history that can be

[1] Unless explicitly required the dependence on the model parameters is dropped for simplicity of notation.
[2] For simplicity of notation the "end-effects" are ignored.

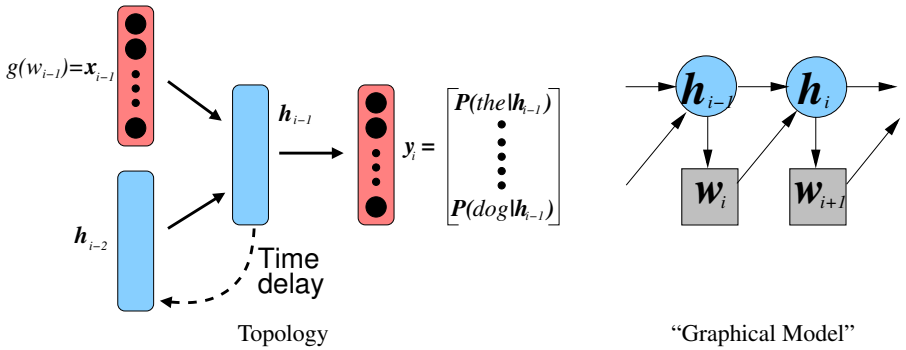


Figure 1.1 *Recurrent Neural Network Language Model*

used in predicting the word. To address this problem *recurrent neural network* language models (Mikolov et al. 2010) have been proposed. A simple illustration of this form of model is shown in Figure 1.1. The left-hand diagram illustrates the topology being used, including "time-delays" for the generation of the history vector. The right-hand diagram illustrates the "graphical model" for this form of model. The observed word sequence $\boldsymbol{w}_{1:L}$ is illustrated as shaded discrete variables, to illustrate that they are discrete and observed. Continuous variables will be shown as circles, and unshaded if they are not observed. The hidden variables, $\boldsymbol{h}_i$ are shaded blue to indicate that they are a *deterministic* function of the variables connected to that node [3]

For this form of model, rather than using a fixed history, a *history vector* $\boldsymbol{h}_i$ is introduced into the model. This is meant to encode all the relevant information from the complete word sequence up to word i. Thus

$$\boldsymbol{h}_i = \boldsymbol{\phi}(\boldsymbol{w}_{1:i}) \tag{1.2}$$

$\boldsymbol{h}_i$ is a fixed length vector generated from the $i-1$ length word sequence $\boldsymbol{w}_{1:i-1}$. This is similar to the concept of the feature space for sequence kernels discussed in more detail in section 1.5. Now

$$P(\boldsymbol{w}_{1:L}) = \prod_{i=1}^{L} P(w_i|\boldsymbol{w}_{1:i-1}) \approx \prod_{i=1}^{L} P(w_i|w_{i-1}, \boldsymbol{h}_{i-2}) \approx \prod_{i=1}^{L} P(w_i|\boldsymbol{h}_{i-1}) \tag{1.3}$$

For this model rather than using a standard sequence kernel feature-space, in the RNNLM this fixed length has the following form is used

$$\boldsymbol{h}_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{h}_{i-1}) = \mathbf{f}^{\mathrm{h}}\left(\mathbf{W}_{\mathrm{h}}^{\mathrm{f}}\boldsymbol{x}_i + \mathbf{W}_{\mathrm{h}}^{\mathrm{r}}\boldsymbol{h}_{i-1} + \boldsymbol{b}_{\mathrm{h}}\right) \tag{1.4}$$

$\mathbf{f}^{\mathrm{h}}()$ is the activation function of the hidden layer. There are more complicated forms possible for $\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{h}_{i-1})$, for example long-short term memory (LSTM) models (Hochreiter & Schmidhuber 1997), and gated recurrent unit (GRU) models (Chung, Gulcehre, Cho

[3] For graphical modes this relationship should be probabilistic. Though deterministic the nodes are shown as they form an integral part of this form of model, and related models.

& Bengio 2014). Additionally it is possible to stack models and add "highway" connections (Irie, Tüske, Alkhouli, Schlüter & Ney 2016). For all these models, the language model probabilities are then obtained from the output layer using

$$P(w_i|\boldsymbol{w}_{1:i-1}) \approx y_{f(w_i)} \tag{1.5}$$

$$\boldsymbol{y}_i = \mathbf{f}^{\mathrm{f}}\left(\mathbf{W}_{\mathrm{y}}\boldsymbol{h}_{i-1} + \boldsymbol{b}_{\mathrm{y}}\right) \tag{1.6}$$

where

- $f(w_i)$ is element in the *1-of-K* encoding of word w_i;
- $\boldsymbol{y}_i$ is the probability mass function of the *1-of-K* encoding of the word at position i in the sequence;
- $\boldsymbol{x}_i = g(w_i)$ is a *1-of-K* encoding of the word at position i, w_i, in the sequence.

The training of these forms of model, as well as the issues, and possible solutions, to including these forms of model in to a speech recognition system, is discussed in more detail in (Liu, Chen, Wang, Gales & Woodland 2016, Chen, Liu, Wang, Gales & Woodland 2016).

1.2 Speech Processing from a Machine Learning Perspective

Speech recognition and speech synthesis can be posed as examples of classification and regression. Speech recognition is a classification task mapping the digitised, τ length, waveform $\boldsymbol{y}_{1:\tau}^{\mathtt{wav}}$ into a word-sequence, $\boldsymbol{w}$. In practice this is often done as a two stage process [4], first performing feature extraction on the waveform, to form a T length observation sequence, $\boldsymbol{Y}_{1:T} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\}$, and then performing classification:

$$\boldsymbol{y}_{1:\tau}^{\mathtt{wav}} \rightarrow \boldsymbol{Y}_{1:T} \rightarrow \boldsymbol{w} \tag{1.7}$$

Note the length of the word-sequence is unknown, but will be less than the length of the observation sequence. Classification is usually an application of Bayes' decision rule. For Bayes' decision rule the classification criterion can be expressed as

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{P(\boldsymbol{w}|\boldsymbol{Y}_{1:T})\} \tag{1.8}$$

assuming that there is a deterministic mapping from the waveform to the feature vectors. This criteria yields the most likely words sequence, sentence. It is possible to use other decoding approaches such as minimum Bayes' risk decoding that yield word sequences that minimise the word error rate. These have the form

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left\{ \sum_{\tilde{\boldsymbol{w}}} P(\tilde{\boldsymbol{w}}|\boldsymbol{Y}_{1:T}) \mathcal{L}(\tilde{\boldsymbol{w}}, \boldsymbol{w}) \right\} \tag{1.9}$$

where $\tilde{\boldsymbol{w}}, \boldsymbol{w})$ is the "loss", measured at for example the word level, between the "hypothesis" $\tilde{\boldsymbol{w}}$ and the "reference" $\boldsymbol{w}$.

[4] There has been increasing interest in integrating the feature extracting stage into the neural network.

Conversely speech synthesis can be viewed as a regression task to obtain the waveform, $\boldsymbol{y}^{\mathtt{wav}}$, from a known word sequence $\boldsymbol{w}_{1:L}$. Again this is commonly expressed as a two stage process [5], of first generating a *trajectory* of features, $\boldsymbol{Y}$, that are then mapped to the waveform:

$$\boldsymbol{w}_{1:L} \rightarrow \boldsymbol{Y} \rightarrow \boldsymbol{y}^{\mathtt{wav}} \tag{1.10}$$

where $\boldsymbol{Y}$ are the set of feature vectors generated at a fixed rate that determine the nature of the speech signal. This process is often simplified to the generation of a set of feature vectors and then the generation of the waveform given these features. Thus synthesis is often written as

$$\hat{\boldsymbol{y}}^{\mathtt{wav}} = \arg\max_{\boldsymbol{y}^{\mathtt{wav}}} \left\{ p(\boldsymbol{y}^{\mathtt{wav}} | \hat{\boldsymbol{Y}}) \right\} \tag{1.11}$$

where the feature sequence (vocoded data) is extracted using

$$\hat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} \{ p(\boldsymbol{Y} | \boldsymbol{w}_{1:L}) \} \tag{1.12}$$

Again the length of the feature vector trajectory will not be same as the length of the word sequence, though the length of the waveform is usually deterministic given the length of the feature vector trajectory.

It is clear that at the heart of both recognition and synthesis is the ability to handle distributions over observation and word sequences that are not of the same length. This *sequence-to-sequence* problem is more challenging than the language modelling example given in the previous section. Central to this process will be the concept of conditional independence assumptions, as used in the language model in the previous section.

1.3 Graphical Models and Conditional Independence

The previous section has discussed the use of sequence models as being at the center of both speech recognition and synthesis. It is interesting to look at possible forms of conditional independence assumptions that will be the underlying assumptions for the sequence-to-sequence models described later in this report [6].

Graphical models (Lauritzen 1996) are a method to describe the conditional independence assumptions in a system. This is a rich research area in its own right, extending far beyond the scope of this book. This section briefly describes some of the most commonly used forms of graphical model in speech processing.

One of the most commonly used models in speech processing and a range of other sequence modelling applications is the *Hidden Markov Model* (HMM) (Gales & Young

[5] As with speech recognition there is increasing interest in "waveform" synthesis.

[6] A trivial generalisation is to consider fix spans of the observation, for example $\boldsymbol{y}_{t-1}, \boldsymbol{y}_t, \boldsymbol{y}_{t+1}$ as the effective observation at time t. This is commonly used in speech processing, but does not alter the underlying modelling process. Note this section uses an equality, as the probabilities are being obtained under the modelling assumptions, rather than expressing approximations to the probabilities.

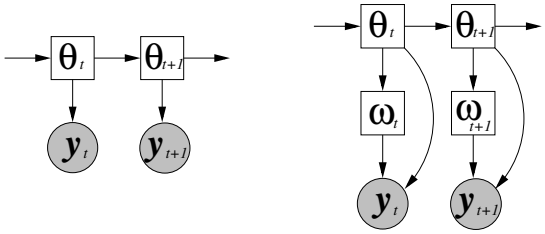


Figure 1.2 *Graphical Model for a Hidden Markov Model: A Dynamic Bayesian Network*

2007). The graphical mode for which is shown in the left in Figure 1.2. This encodes the following conditional independence assumptions

1. *states*: state transition distributions are conditional independent given the previous state. Thus

$$P(\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} P(\theta_t|\theta_{t-1}) \tag{1.13}$$

2. *observations*: state output distributions are conditionally independent given the state. Thus

$$p(\mathbf{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\theta_t) \tag{1.14}$$

   For the combination of the HMM and GMM, shown on the right of Figure 1.2 the state output distribution at time t is also a function of the unobserved, discrete component, ω_t. Thus

$$p(\mathbf{Y}_{1:T}|\boldsymbol{\theta}_{1:T}, \boldsymbol{\omega}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\theta_t, \omega_t) \tag{1.15}$$

   The state output probability is obtained by marginalising out over the components

$$p(\mathbf{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} \sum_{\omega \in \{\boldsymbol{\Omega}\}_T} P(\omega|\theta_t) p(\boldsymbol{y}_t|\theta_t, \omega) \tag{1.16}$$

The overall likelihood of the model generating the observation sequence can then be expressed as

$$p(\mathbf{Y}_{1:T}) = \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} \prod_{t=1}^{T} p(\boldsymbol{y}_t|\theta_t) P(\theta_t|\theta_{t-1})$$

where $\{\boldsymbol{\Theta}\}_T$ is the set of all possible T-length state sequences [7].

[7] At this stage there is no connection with the word sequence. To add the likelihood that the word-sequence generated the observation sequence the set of all possible state sequence is restricted to those associated with a particular word sequence.

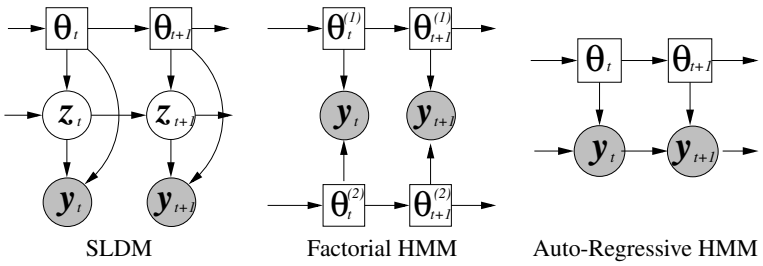


Figure 1.3 *Dynamic Bayesian Networks for Switching Linear Dynamical Model (left), Factorial HMM (center) and Auto-Regressive HMM (right)*

Other forms of graphical models have also been applied in the speech processing area. For example additional, continuous, latent variables can be added (Roweiss & Ghahramani 1999, Rosti & Gales 2001) yielding *Switching Linear Dynamical Models* (SLDMs), or additional discrete latent variables for *Factorial HMMs*. Finally the state distribution can be made a function of the previous observation, a *Auto-Regressive HMM* (ARHMM). All of these modified graphical models are shown in Figure 1.3. Based on the form of graphical model, which defines the conditional independence assumptions, the probability of the observation sequence can be derived. For example in the case of the factorial HMMs

$$p(\mathbf{Y}_{1:T}) = \sum_{\boldsymbol{\theta}^{(1)}_{1:T} \in \{\boldsymbol{\Theta}\}^{(1)}_T} \sum_{\boldsymbol{\theta}^{(2)}_{1:T} \in \{\boldsymbol{\Theta}\}^{(2)}_T} \prod_{t=1}^{T} p(\boldsymbol{y}_t|\theta^{(1)}_t, \theta^{(2)}_t) P(\theta^{(1)}_t|\theta^{(1)}_{t-1}) P(\theta^{(2)}_t|\theta^{(2)}_{t-1}) \tag{1.17}$$

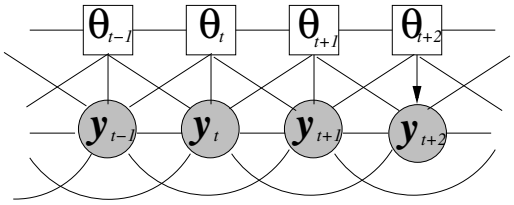


Figure 1.4 *Trajectory HMM: An Undirected Graphical Model*

The examples of graphical models given above are all *Bayesian Networks*, Directed Acyclic Graphs (DAGs). Other forms of graphical model are also possible. One form that has been used for both speech recognition and synthesis is the *undirected* graphical model. These can be used to represent *product of experts* (Zen, Gales, Nankaku & Tokuda 2012). One example of this is the trajectory HMM (Zen, Tokuda & Kitamura 2007), which is used in speech synthesis. The graphical model for this is shown in Figure 1.4. This may be viewed as a *globally normalised* model, rather than the form of local normalisation (to yield a probability) that is used for BNs. The likelihood for the

model in Figure 1.4 can be written as

$$p(\boldsymbol{Y}_{1:T}) = \frac{1}{Z} \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} \prod_{t=1}^{T} p\left(\boldsymbol{f}(\boldsymbol{Y}_{t-2:t+2}, \boldsymbol{\theta}_{t-1:t+1})\right) \tag{1.18}$$

where $\boldsymbol{f}(\boldsymbol{Y}_{t-2:t+2}, \boldsymbol{\theta}_{t-1:t+1})$ is a feature-vector function of both the observation sequence and the state sequence.

As discussed in section 1.1, in addition to the descriptions of the standard graphical models, an additional deterministic node will be added in this work, see Figure 1.1. Though this can be subsumed within connecting nodes they will be kept distinct in this report since it allows the dependencies of the neural network nodes to be explicitly described.

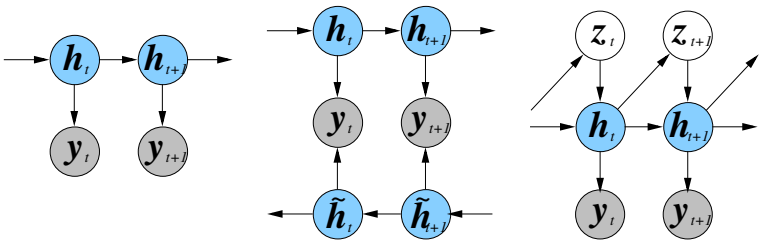


Figure 1.5 *Three variants of recurrent neural networks*

Figure 1.5 shows three variants of RNN. Note that both GRUs and LSTMs have the same form in terms of dependencies. The deterministic nodes are shown in blue. For the RNN model shown above

$$p(\boldsymbol{Y}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t | \boldsymbol{h}_t) \tag{1.19}$$

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{h}_{t-1}, \boldsymbol{y}_{t-1}) \tag{1.20}$$

As described for the RNNLM, the history vector can model a representation of the complete history of observation vectors. Two modifications to the baseline RNN are also shown. The first version is the bi-directional RNN (Schuster & Paliwal 1997) where the following approximation is used

$$p(\boldsymbol{y}_t | \boldsymbol{Y}_{1:t-1}, \boldsymbol{Y}_{t+1:T}) = p(\boldsymbol{y}_t | \boldsymbol{h}_t, \tilde{\boldsymbol{h}}_t) \tag{1.21}$$

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{h}_{t-1}, \boldsymbol{y}_{t-1}) \tag{1.22}$$

$$\tilde{\boldsymbol{h}}_t = \boldsymbol{f}(\boldsymbol{h}_{t+1}, \boldsymbol{y}_{t+1}) \tag{1.23}$$

Care must be taken with this form of model, as the probability of the observation at a particular time instance is a function of both the previous and future observations via the history vectors $\boldsymbol{h}_t$ and $\tilde{\boldsymbol{h}}_t$ respectively. The probability of the sequence can be expressed

as

$$p(\boldsymbol{Y}_{1:T}) = \frac{1}{Z}\prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{h}_t, \tilde{\boldsymbol{h}}_t) \tag{1.24}$$

where the normalisation term ensures that this is valid PDF. A second modification is to introduce latent variables $\boldsymbol{Z}_{1:T} = \boldsymbol{z}_1, \ldots, \boldsymbol{z}_T$. The probability of the observation sequence for this latent variable RNN (Chung, Kastner, Dinh, Goel, Courville & Bengio 2015) can be expressed as

$$p(\boldsymbol{Y}_{1:T}) = \int \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{h}_t)p(\boldsymbol{z}_t|\boldsymbol{h}_{t-1})d\boldsymbol{Z}_{1:T} \tag{1.25}$$

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{h}_{t-1}, \boldsymbol{z}_t, \boldsymbol{y}_{t-1}) \tag{1.26}$$

It is interesting to consider the forms of temporal conditional independence that these probabilistic and deterministic relationships between variables can describe. This has important implications for both training and inference.

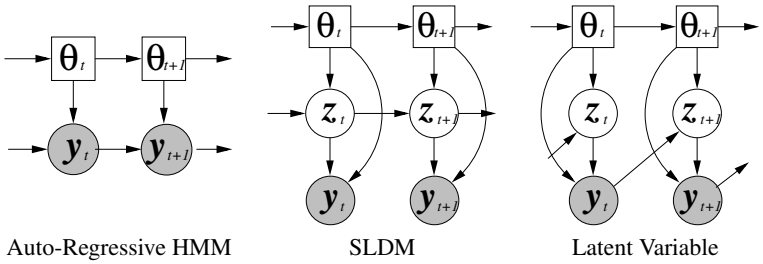


Figure 1.6 *Dynamic Bayesian Networks for Switching Linear Dynamical Model (left), Factorial HMM (center) and Auto-Regressive HMM (right)*

Figure 1.6 shows three forms of DBN that will be used to illustrate the temporal modelling. Given the state sequence, $\boldsymbol{\theta}_{1:T}$, the probabilities is discussed below.

- Auto-Regressive HMM (AR-HMM): observation dependence extension of the HMM, Figure 1.2. Here

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}, \boldsymbol{\theta}_t) \tag{1.27}$$

  The most common form of distribution is a Gaussian. Thus

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}, \boldsymbol{\theta}_t) = \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{A}^{(\mathrm{y})}_{\theta_t}\boldsymbol{y}_{t-1} + \boldsymbol{\mu}_{\theta_t}, \boldsymbol{\Sigma}_{\theta_t}) \tag{1.28}$$

  where $\boldsymbol{A}^{(\mathrm{y})}_{\theta_t}$ is the linear transform representing the impact of the previous observation on the mean of the current observation.

- Switching Linear Dynamic Model (SLDM):

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \int p(\boldsymbol{Y}_{1:t}|\boldsymbol{Z}_{1:T}, \boldsymbol{\theta}_{1:T}) p(\boldsymbol{Z}_{1:T}|\boldsymbol{\theta}_{1:T}) d\boldsymbol{Z}_{1:T} \tag{1.29}$$

$$= \int \left(\prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{z}_t, \theta_t) p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \theta_t) \right) d\boldsymbol{Z}_{1:T} \tag{1.30}$$

  The standard form for this expression uses Gaussian distribution and linear relationships between the variables. Thus

$$p(\boldsymbol{y}_t|\boldsymbol{z}_t, \theta_t) = \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}_{\theta_t} + \boldsymbol{C}_{\theta_t}\boldsymbol{z}_t, \boldsymbol{\Sigma}_{\theta_t}) \tag{1.31}$$

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{\theta}_t) = \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{A}^{(\mathrm{z})}_{\theta_t}\boldsymbol{z}_{t-1} + \boldsymbol{\mu}^{(\mathrm{z})}_{\theta_t}; \boldsymbol{\Sigma}^{(\mathrm{z})}_{\theta_t}) \tag{1.32}$$

  The latent variable $\boldsymbol{z}_t$ is this model yields a continuous representation of the complete history of unobserved states $\boldsymbol{\theta}_{1:t}$.

- Latent Variable Model: introduce a latent variable as a function of the previous observation

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \int p(\boldsymbol{Y}_{1:t}|\boldsymbol{Z}_{1:T}, \boldsymbol{\theta}_{1:T}) p(\boldsymbol{Z}_{1:T}) d\boldsymbol{Z}_{1:T} \tag{1.33}$$

$$= \int \left(\prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{z}_t, \theta_t) p(\boldsymbol{z}_t|\boldsymbol{y}_{t-1}, \boldsymbol{z}_{t-1}) \right) d\boldsymbol{Z}_{1:T} \tag{1.34}$$

  The simplest form of dependency for the latent variable is to make it Gaussian

$$p(\boldsymbol{z}_t|\boldsymbol{y}_{t-1}, \boldsymbol{z}_{t-1}) = \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{A}^{(\mathrm{y})}\boldsymbol{y}_{t-1} + \boldsymbol{A}^{(\mathrm{z})}\boldsymbol{z}_{t-1} + \boldsymbol{\mu}^{(\mathrm{z})}; \boldsymbol{\Sigma}^{(\mathrm{z})}) \tag{1.35}$$

  Here the latent variable $\boldsymbol{z}_t$ can be viewed as a continuous representation of the previous observations $\boldsymbol{Y}_{1:t-1}$

In addition, for all models illustrated in Figure 1.6 the probability of the state-sequence given the word-sequence can be expressed as

$$P(\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} P(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) \tag{1.36}$$

The nature of these temporal dependencies will be partitioned into three broad groups.

1. *Continuous/discrete observed*: this the form of dependency shown in the AR-HMM between the continuous observations, in addition there is the dependence on the state. Normally the observations that are extracted only contain information about the current window, they are not designed to contain information about the back-history of observations. This dramatically limits the ability of this form of model to represent the history.
2. *Discrete unobserved*: this the standard discrete Markov assumption. The discrete state contains sufficient information for the complete history. In practice, without an exponential increase in the number of discrete states, this can only be used to

represent a finite history [8]. In Figure 1.6 the state-sequence $\boldsymbol{\theta}_{1:T}$ is assumed to be generated under this assumption.

3. *Continuous unobserved*: this is the form of latent variable $\boldsymbol{z}_t$ in the SLDM and latent variable model. Compared to the observed continuous variables there is the ability to both decide the dimensionality of the system, and to allow the system to learn how the latent variable parameters should be used.

The discussion above has focused on standard graphical models where there are probabilistic relationships between all the variables. The same concepts can be applied to the deterministic relationships described in the discussion of RNN variants, Figure 1.5. There are two interesting differences between the standard, linear Gaussian, distribution discussed above and the form for RNNs.

1.4 Generative Sequence-to-Sequence Models

For many years the dominant form of model used in speech recognition and speech synthesis was based on *generative models*, in particular forms of Hidden Markov Models (HMMs) (Gales & Young 2007). These systems model the joint distribution of the observations and the labels. This joint distribution is then marginalised out to yield the distribution of interest. This section, and the next section, will focus on speech recognition, though similar forms of analysis can be done for speech synthesis.

The majority of speech recognition systems handle the difference in length between the L-length word sequence, $\boldsymbol{w}_{1:L}$ and the T-length observation sequence $\boldsymbol{Y}_{1:T}$ by introducing discrete latent variables, states, $\boldsymbol{\theta}_{1:T}$. The joint distribution is then expressed as

$$p(\boldsymbol{Y}_{1:T}, \boldsymbol{w}_{1:L}) = P(\boldsymbol{w}_{1:L}) \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} p(\boldsymbol{Y}_{1:T}, \boldsymbol{\theta}_{1:T} | \boldsymbol{w}_{1:L}) \tag{1.37}$$

$$= P(\boldsymbol{w}_{1:L}) \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} P(\boldsymbol{\theta}_{1:T} | \boldsymbol{w}_{1:L}) p(\boldsymbol{Y}_{1:T} | \boldsymbol{\theta}_{1:T}) \tag{1.38}$$

where $\{\boldsymbol{\Theta}\}_T$ is the set of all T-length state sequences. This has now yielded three distinct models:

- *language model* $P(\boldsymbol{w}_{1:L})$: this was briefly discussed at the start of this report;
- *alignment model* $P(\boldsymbol{\theta}_{1:T} | \boldsymbol{w}_{1:L})$: the "relationship" between the unobserved state sequence and the word sequence;
- *acoustic model* $p(\boldsymbol{Y}_{1:T} | \boldsymbol{\theta}_{1:T})$: the probability of the observation sequence given the state-sequence.

The nature of the alignment model and acoustic model will now be described in more detail.

[8] This is a slight simplification as cache and trigger forms of model can also be used. These allow longer span dependencies to be incorporated, but empirically have been found to yield limited gains.

Classification with this form of model relies on Bayes' rule. Thus for an observed T-length observation sequence $\boldsymbol{Y}_{1:T}$ Eqn 1.8 is expressed as

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{P(\boldsymbol{w}|\boldsymbol{Y}_{1:T})\} = \arg\max_{\boldsymbol{w}} \{P(\boldsymbol{Y}_{1:T}|\boldsymbol{w})P(\boldsymbol{w})\} \tag{1.39}$$

1.4.1 Alignment Model

The alignment model associated with speech recognition system is itself normally split into two models. The first is the mapping from the word sequences to sub-word models, the *lexicon*, $P(\boldsymbol{s}|\boldsymbol{w}_{1:L})$. The second model is the *duration model* which yields the probability of a particular allocation of the T length feature vector into sub-word models, $P(\boldsymbol{\theta}_{1:T}|\boldsymbol{s})$. So

$$P(\boldsymbol{\theta}_{1:T}|\boldsymbol{w}_{1:L}) = \sum_{\boldsymbol{s}\in\{\mathcal{S}\}} P(\boldsymbol{\theta}_{1:T}|\boldsymbol{s})P(\boldsymbol{s}|\boldsymbol{w}_{1:L}) \tag{1.40}$$

where $\{\mathcal{S}\}$ is the set of all sub-word sequences associated with the word-sequence $\boldsymbol{w}_{1:L}$. The base form of the lexicon is usually derived by hand, or using a small set of pronunciations and a *grapheme to phoneme* (G2P) model (Bisani & Ney 2008). Pronunciation probabilities can then be added to this lexicon, yielding a standard word by word partition of word in sub-word unit.

The second model is itself a doubly sequential model connecting the T-length observation sequence with a K-length state. The two sequences are linked by assigning durations to the K-length state sequence. This is usually modelled as Markov process with an associate duration model. Thus

$$P(\boldsymbol{\theta}_{1:T}|\boldsymbol{s}_{1:K}) = \prod_{i=1}^{K} P(d(\mathsf{s}_i|\boldsymbol{\theta}_{1:T}, i)|\mathsf{s}_i) \tag{1.41}$$

where $d(\mathsf{s}_i|\boldsymbol{\theta}_{1:T}, i)$ yields the number of frames allocated to state s_i (at occurrence i) in the state-sequence $\boldsymbol{\theta}_{1:T}$. The simplest form of model, used in standard HMMs, is to state

$$P(d(\mathsf{s}_i|\boldsymbol{\theta}_{1:T}, i)|\mathsf{s}_i) = P(d_i|\mathsf{s}_i) = \prod_{\tau=1}(1 - a_{ii}^{\mathsf{s}_i})(a_{ii}^{\mathsf{s}_i})^{d_i} \tag{1.42}$$

where $a_{ii}^{\mathsf{s}_i}$ is the self-loop probability for state i.

1.4.2 Acoustic Model

A range of acoustic models, that yield $p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T})$, have been examined for speech processing (Gales & Young 2007, Hinton, Deng, Yu, Dahl, Mohamed, Jaitly, Senior, Vanhoucke, Nguyen, Sainath & Kingsbury 2012, Schuster & Paliwal 1997, Robinson & Fallside 1991). At the heart of these models is the nature of the conditional independence assumptions discussed in section 1.3. There are two T-length sequences that determine the likelihood: the observed feature vectors $\boldsymbol{Y}_{1:T}$; and the state sequence $\boldsymbol{\theta}_{1:T}$. Dependencies within both sequences need to be considered.

In this report the nature of the dependencies will be described as*Markovian* where a fixed-length "history" is being used, or *non-Markovian* where a complete history, or

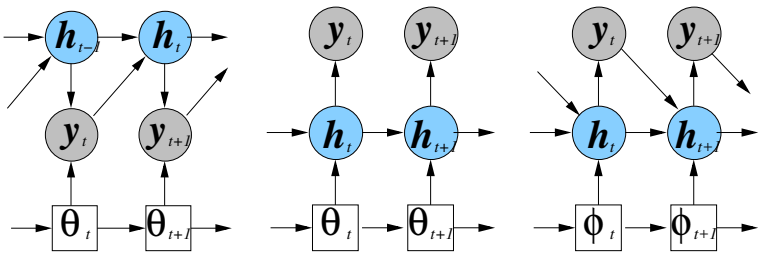


Figure 1.7 *Dynamic Bayesian three forms of acoustic model dependencies*

representation thereof, is used. Figure 1.7 shows three dependencies for three of the models described.

1. **Fully Markovian**: this is the simplest form of model. Here

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\theta_t) \tag{1.43}$$

   The observations are conditionally independent given the state at that time instance. This expression is the underlying acoustic model for *Hidden Markov Models* (HMMs), the BN for this model was shown in Figure 1.2. As previously discussed it is possible to extend the dependencies to fixed numbers of previous states, or features. For example if only a fixed history of observation features are considered, in this case the previous observation, then

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}, \theta_t) \tag{1.44}$$

   yielding an *Auto-regressive HMM*, see Figure 1.6.
2. **State Markovian**: here the observations not only depend on the current state, but also on the all previous observations. Thus the feature dependencies are non-Markovian, whereas the state-dependencies are Markovian. This is illustrated in Figure 1.7(a). The the conditional likelihood can be expressed as

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{Y}_{1:t-1}, \theta_t) \tag{1.45}$$

   It is impractical to use this precise form, as there would be an exponential growth in the number of model parameters. Instead of trying to explicitly model the complete history of observations, a compact fixed-dimensional representation of this history can be used. Thus

$$p(\boldsymbol{y}_t|\boldsymbol{Y}_{1:t-1}, \theta_t) \approx p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1}, \theta_t) \tag{1.46}$$

   where $\boldsymbol{h}_{t-1}$ is a fixed dimension representation of the observation sequence $\boldsymbol{Y}_{1:t-1}$

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{y}_{t-1}, \boldsymbol{h}_{t-1}) \tag{1.47}$$

3. **Feature Markovian**: here the generated features are assumed to be conditionally independent given the current state, illustrated in Figure 1.7(b). Thus

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t}) \tag{1.48}$$

   This has not directly addressed the problem of the growth in the number of model parameters. Instead of trying to explicitly model the complete history if observations, a compact fixed-dimensional representation of this history is used. Here

$$p(\boldsymbol{y}_t|\boldsymbol{\theta}_{1:t}) \approx p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1}, \theta_t) \approx p(\boldsymbol{y}_t|\boldsymbol{h}_t) \tag{1.49}$$

   where $\boldsymbol{h}_{t-1}$ is a fixed dimension representation of the state sequence $\boldsymbol{\theta}_{1:t-1}$

$$\boldsymbol{h}_t = \boldsymbol{f}(\theta_t, \boldsymbol{h}_{t-1}) \tag{1.50}$$

   This is the basis of the *Recurrent Neural Network* acoustic models in speech synthesis (Zen & Sak 2015).
4. **Non-Markovian**: here as much information as possible is used.

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{Y}_{1:t-1}, \boldsymbol{\theta}_{1:t}) \tag{1.51}$$

   Again a history vector can be used to yield a compact representation of this information

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{y}_{t-1}, \theta_t, \boldsymbol{h}_{t-1}) \tag{1.52}$$

One of the issues that arises from using feature Markovian, or non-Markovian, forms of model for speech recognition is that there is a dependence on an unobserved latent variable. This results in a dependence e on a complete back-history of states. This is exactly the same problem as using an RNNLM during decoding.

An interesting question when using deep neural networks in generative models is the form of network, and training criterion. It is possible to use Mixture Density Neural Networks (Bishop 1994) to yield a probability density function, often by using the network to predict the parameters of a standard model such as a Gaussian distribution. This form has been used for speech synthesis (Zen & Senior 2014). For speech recognition within a generative modelling framework, it is more common to train a discriminative model and then convert the output into a *pseudo likelihood*. Consider the fully Markovian case, the HMM-based model. Here

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) \approx \prod_{t=1}^{T} p(\boldsymbol{y}_t|\theta_t) = \prod_{t=1}^{T} \frac{P(\theta_t|\boldsymbol{y}_t)p(\boldsymbol{y}_t)}{P(\theta_t)} \tag{1.53}$$

$P(\theta_t|\boldsymbol{y}_t$ can be estimated as a standard discriminative DNN, $p(\boldsymbol{y}_t)$ does not depend on the word-sequence, and $P(\theta_t)$ is the prior of the state.

It is more interesting (challenging) to consider the state Markovian case from Figure 1.7(a). The acoustic model component can be expressed as

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) \approx \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1}, \theta_t) = \prod_{t=1}^{T} \frac{P(\theta_t|\boldsymbol{y}_t, \boldsymbol{h}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1})}{P(\theta_t|\boldsymbol{h}_{t-1})} \tag{1.54}$$

The first numerator term, $P(\theta_t|\boldsymbol{y}_t, \boldsymbol{h}_{t-1})$, is a discriminative RNN and will be discussed in section 1.5. The second numerator, $p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1})$ is only a function of the observation sequence. However, tough its value does not depend on the word-sequence, it does depend on the current model parameters through the value of the history vector $\boldsymbol{h}_{t-1}$. The denominator term, $P(\theta_t|\boldsymbol{h}_{t-1})$, is problematic as it is both a function of the model parameters and the state [9]. To address this problem an additional approximation is made, that the $P(\theta_t|\boldsymbol{h}_{t-1}) \approx P(\theta)$. Thus

$$p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T}) \approx \prod_{t=1}^{T} \frac{P(\theta_t|\boldsymbol{y}_t, \boldsymbol{h}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1})}{P(\theta_t)} \propto \prod_{t=1}^{T} \frac{P(\theta_t|\boldsymbol{y}_t, \boldsymbol{h}_{t-1})}{P(\theta_t)} \tag{1.57}$$

This form of pseudo-likelihood is the standard form that is used in hybrid systems (Bourlard & Morgan 1994).

1.5 Discriminative Sequence-to-Sequence Models

The previous section has focused on generative models. This is natural to do for speech synthesis, a fundamentally generative process, but not so obvious for speech recognition which is a classification, discrimination, task. It is therefore of interest to examine how discriminative models (Bishop 2006) can be applied to sequence-to-sequence modelling tasks such as speech recognition. In a discriminative model the posterior probability are directly modelled, rather than obtained via Bayes rule. Discriminative models have the same issue as generative models: how to prevent an exponential growth in the number of model parameters. The issue is addressed in a similar fashion by making conditional independence assumptions and compact representations of "histories". Additionally latent variables can be introduced to handle differences in the length of the sequences. For discriminative models the introduction of the latent state sequence $\boldsymbol{\theta}_{1:T}$ yields

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} P(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) \tag{1.58}$$

$$\approx \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} P(\boldsymbol{w}_{1:L}|\boldsymbol{\theta}_{1:T})P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) \tag{1.59}$$

[9] The correct approach given the model would be to compute

$$P(\theta_t|\boldsymbol{h}_{t-1}) = \int P(\theta_t|\boldsymbol{y}, \boldsymbol{h}_{t-1})p(\boldsymbol{y}|\boldsymbol{h}_{t-1})d\boldsymbol{y} \tag{1.55}$$

Note for the fully-Markovian case the standard approximation is reasonable as

$$P(\mathsf{s}_i) = \int P(\mathsf{s}_i|\boldsymbol{y})p(\boldsymbol{y})d\boldsymbol{y} \approx \frac{1}{T}\sum_{t=1}^{T} P(\theta_t = \mathsf{s}_i|\boldsymbol{y}_t) \tag{1.56}$$

An alternative way, related to sequence kernels, is to use a fix-length encoding of the observation sequence

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) \approx \prod_{i=1}^{L} P(w_i|\phi(\boldsymbol{Y}_{1:T}, \boldsymbol{w}_{1:i-1})) \tag{1.60}$$

This section will briefly describe the following four models based on the above expressions:

- *discriminative alignment model* $P(\boldsymbol{w}_{1:L}|\boldsymbol{\theta}_{1:T}; \boldsymbol{\lambda})$: this is the mapping of the state-sequence to the word-sequence;
- *discriminative acoustic model* $P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}; \boldsymbol{\lambda})$: the mapping from the observations to the state sequence;
- *structured discriminative model* $P(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}; \boldsymbol{\lambda})$: the joint distribution is modelled and the posterior obtained by marginalising;
- *encoder-decoder model*: $P(w_i|\phi(\boldsymbol{Y}_{1:T}, \boldsymbol{w}_{1:i-1}))$: where the nature of the mapping of variable length sequence to fixed (or the length of the word-sequence), $\phi(\boldsymbol{Y}_{1:T}, \boldsymbol{w}_{1:i-1})$, will be discussed.

1.5.1 Discriminative Alignment Models

These models have the same operation as the generative model alignment models connecting the L-length word sequence and the T-length observations sequence. If the system is configured so that the lexicon defines a unique mapping from a state sequence to word sequence [10], then

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{\theta}_{1:T}) = 1 \tag{1.61}$$

This then yields the following expression

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) \tag{1.62}$$

An interesting aspect of this form of model is that all the history information, for both the state sequence and observation sequence must be contained within the discriminative acoustic model. This is more challenging than the generative model where the word sequence is modelled separately (by the language model). The language model only requires text training data, which odes not have to be in domain or have associated acoustic data. Additionally the length of the state sequence is often significantly longer than the word sequence, $T > L$.

The simplest option to address these issues is to combine the discriminative acoustic model with a language score within a log-linear model framework. Thus

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \frac{1}{Z(\boldsymbol{Y}_{1:T})} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}} \begin{bmatrix} \log\left(\sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T})\right) \\ \log(\tilde{P}(\boldsymbol{w}_{1:L})) \end{bmatrix}\right) \tag{1.63}$$

where the normalisation term $Z(\boldsymbol{Y}_{1:T})$ ensures that this is a valid PMF and is a function

[10] This has ignored hompohones for phonetic lexicons and homographes for graphemic (character) lexicons.

of the observation sequence $\boldsymbol{Y}_{1:T}$. This term only impacts the training of the model, not inference. Note $\tilde{P}(\boldsymbol{w}_{1:L})$ is used to denote the language model, rather than $P(\boldsymbol{w}_{1:L})$. This is to emphasis the fact that the model can be trained on additional data and will not necessarily be derivable, for example, from the final discriminative model.

Extensions beyond the simple, shallow combination, above is to separately train the networks, one on the available language model data the other on the acoustic data, and then combine using an additional network (Graves 2012). This is based on modelling the joint distribution of the word and state sequence [11]

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) \approx \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} \prod_{i=1}^{L} P(w_i, \boldsymbol{\theta}_{\tau^{(i-1)}+1:\tau^{(i)}}|\boldsymbol{w}_{1:i-1}, \boldsymbol{\theta}_{1:\tau^{(i-1)}}, \boldsymbol{Y}_{1:T}) \tag{1.64}$$

where the joint distribution has been written in a conditional form with word w_i and word end-time $\tau^{(i)}$ specified by the state sequence $\boldsymbol{\theta}_{1:T}$. The conditional probability is now expressed as a product of (separately trained) recurrent neural network experts

$$\begin{aligned} P(w_i, \boldsymbol{\theta}_{\tau^{(i-1)}+1:\tau^{(i)}}|\boldsymbol{w}_{1:i-1}, \boldsymbol{\theta}_{1:\tau^{(i-1)}}, \boldsymbol{Y}_{1:T}) &= \frac{1}{Z}\left(\tilde{P}(w_i|\boldsymbol{w}_{1:i-1}) \prod_{t=\tau^{(i-1)}+1}^{\tau^{(i)}} P(\theta_t|, \boldsymbol{\theta}_{1:t-1}, \boldsymbol{Y}_{1:T})\right) \\ &\approx \frac{1}{Z}\left(\tilde{P}(w_i|\tilde{\boldsymbol{h}}_{i-1}) \prod_{t=\tau^{(i-1)}+1}^{\tau^{(i)}} P(\theta_t|, \boldsymbol{h}_{t-1})\right) \\ &\approx \prod_{t=\tau^{(i-1)}+1}^{\tau^{(i)}} P(\theta_t|\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_{t-1}) \end{aligned} \tag{1.65}$$

where $\tilde{\boldsymbol{h}}_i$ and $\boldsymbol{h}_t$ are the hidden units, from the language and acoustic models respectively. This is now an asynchronous factorial search problem jointly over the word sequence and word end-times. The form of probability $P(\theta_t|\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_{t-1})$ can be made relatively simple, relying on the pretrained models to yield discriminative hidden vectors.

1.5.2 Discriminative Acoustic Model

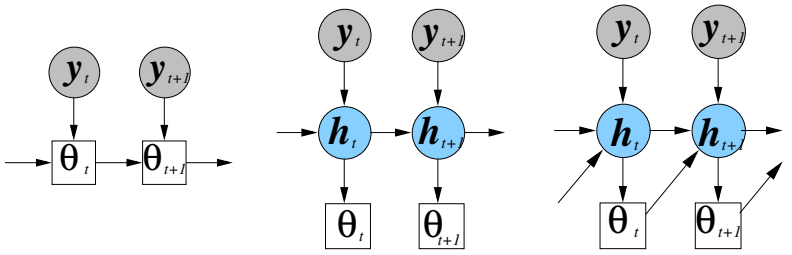
The discriminative acoustic model has to handle the same problems as the generative acoustic model described previously. Again dependencies will be described as Markovian or non-Markovian.

1. **Fully Markovian**: here finite history dependencies are considered for both the state and feature dependencies. Thus

$$P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) = \prod_{t=1}^{T} P(\theta_t|\boldsymbol{y}_t, \theta_{t-1}) \tag{1.66}$$

   This is the basis of the maximum entropy Markov model (MEMM) (Kuo & Gao 2006) shown in Figure 1.8(a).

[11] The approximation results from inter-dependencies of the word and state, on future word and state sequences have been ignored.



(a) Fully Markovian (b) State Markovian RNN (c) Non-Markovian RNN

Figure 1.8 *Dynamic Bayesian Networks for different forms of discriminative models and the nature of the dependencies.*

2. **State Markovian**: again considering the form where the state posterior is dependent on the complete history of observations.

$$P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) = \prod_{t=1}^{T} P(\theta_t|\boldsymbol{Y}_{1:t}) \tag{1.67}$$

Similar to the generative model form, this does not explicitly address the exponential growth in the number of model parameters. An RNN can again be used where

$$P(\theta_t|\boldsymbol{Y}_{1:t-1}) \approx P(\theta_t|\boldsymbol{y}_t, \boldsymbol{h}_{t-1}) \approx P(\theta_t|\boldsymbol{h}_t) \tag{1.68}$$

where

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{y}_t, \boldsymbol{h}_{t-1}) \tag{1.69}$$

This is show in Figure 1.8(b). This is the form of model that is used in Connectionist Temporal Classification (CTC) (Graves, Fernández, Gomez & Schmidhuber 2006).

3. **Non-Markovian**: it is also possible to include all dependencies in the system.

$$P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) = \prod_{t=1}^{T} P(\theta_t|\boldsymbol{Y}_{1:t}, \boldsymbol{\theta}_{1:t-1}) \tag{1.70}$$

Incorporating a history vector, as shown in Figure 1.8(c), restricts the number of model parameters. Here

$$P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) = \prod_{t=1}^{T} P(\theta_t|\boldsymbol{Y}_{1:t}, \boldsymbol{\theta}_{1:t-1}) \approx \prod_{t=1}^{T} P(\theta_t|\boldsymbol{y}_t, \theta_{t-1}, \boldsymbol{h}_{t-1}) \approx \prod_{t=1}^{T} P(\theta_t|\boldsymbol{h}_t) \tag{1.71}$$

where

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{y}_t, \theta_{t-1}, \boldsymbol{h}_{t-1}) \tag{1.72}$$

Thus the vector $\boldsymbol{h}_t$ represents both the state and the state sequence.

It is useful to briefly discuss the impact that the forms of the acoustic model dependencies can have on the complexity of the inference, recognition, process. From

Equation 1.8 speech recognition requires computing

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{P(\boldsymbol{w}|\boldsymbol{Y}_{1:T})\} = \arg\max_{\boldsymbol{w}} \left\{ \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T^{(\boldsymbol{w})}} P(\boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) \right\} \tag{1.73}$$

where $\{\boldsymbol{\Theta}\}_T^{(\boldsymbol{w})}$ is the set of all T-length state sequences that are possible for word sequence $\boldsymbol{w}$. For models that Markovian in the states, for example those shown in Figure 1.8(a)(b), it is possible to use the efficient Viterbi algorithm (Viterbi 1967) to find the optimal word sequence. For the non-Markovian case Figure 1.8(c) it is necessary to compute

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \left\{ \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T^{(\boldsymbol{w})}} \prod_{t=1}^{T} P(\theta_t|\boldsymbol{Y}_{1:t}, \boldsymbol{\theta}_{1:t-1}) \right\} \tag{1.74}$$

In this case it is not possible to use Viterbi as the probability of the state at time t, θ_t, depends on the complete state history up to that time instance $\boldsymbol{\theta}_{1:t-1}$.

1.5.3 Structured Discriminative Models

An alternative to the above form is to jointly model the word sequence, state-sequence and observation sequence. To handle variable length sequences sequence kernels, and their associated score-spaces, are used. These models are sometimes referred to as *structured discriminative models* (Gales, Watanabe & Fosler-Lussier 2012). Two forms of model are possible depending on the nature of the normalisation:

- **globally normalised**:

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \frac{1}{Z(\boldsymbol{Y}_{1:T})} \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}, \boldsymbol{Y}_{1:T})\right) \tag{1.75}$$

  where $\boldsymbol{\phi}(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}, \boldsymbol{Y}_{1:T})$ generates a *score-space* from the word, observation, and Z is the normalisation term.

- **locally normalised**:

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} P(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}|\boldsymbol{Y}_{1:T}) \tag{1.76}$$

$$= \sum_{\boldsymbol{\theta}_{1:T} \in \{\boldsymbol{\Theta}\}_T} \frac{1}{Z(\boldsymbol{Y}_{1:T})} \exp\left(\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}, \boldsymbol{Y}_{1:T})\right) \tag{1.77}$$

Both models make use of a general score-space function, $\boldsymbol{\phi}(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}, \boldsymbol{Y}_{1:T})$. This function makes this a very flexible model. For example one form of score space is related to the generative classifier

$$\boldsymbol{\phi}(\boldsymbol{w}_{1:L}, \boldsymbol{\theta}_{1:T}, \boldsymbol{Y}_{1:T}) = \begin{bmatrix} \log(p(\boldsymbol{Y}_{1:T}|\boldsymbol{\theta}_{1:T})) \\ \log(P(\boldsymbol{\theta}_{1:T}|\boldsymbol{w}_{1:L})) \\ \log(P(\boldsymbol{w}_{1:L})) \end{bmatrix} \tag{1.78}$$

A key issue for these log-linear models, and other models, is to map variable length

sequences of vectors or scalars to a fixed length feature vector (or scalar). For example consider the T-length observation sequence $\boldsymbol{Y}_{1:T}$, where the length T can vary. This is converted to a fixed length vector $\boldsymbol{c}$ for all possible T using

$$\boldsymbol{Y}_{1:T} \rightarrow \phi(\boldsymbol{Y}_{1:T}) = \boldsymbol{c} \tag{1.79}$$

A very simple form of function is shown in Equation 1.78 where the variable length sequence is mapped to a log-probability using a statistical model. Other more complicated forms of mapping have also been proposed for both continuous (Smith & Gales 2002) and discrete (Cortes, Haffner & Mohri 2004). More recently the history vector representation from the RNN (or LSTM), $\boldsymbol{h}_{t-1}$ have also been proposed (Lu, Kong, Dyer, Smith & Renals 2016). These sequence kernels also have close connections to the encoder-decoder models described in the next section.

1.5.4 Encoder-Decoder Networks

An alternative approach is to use an encoder-decoder framework (Sutskever, Vinyals & Le 2014). Originally this was used for machine translation, but the description here will be in terms of speech recognition. A neural network is used to encode the observation sequence $\boldsymbol{Y}_{1:T}$ into fixed length feature vector $\boldsymbol{c}$ in the same fashion as the sequence kernel in the previous section. The probability of the word sequence $\boldsymbol{w}_{1:L}$ is then computed conditioned on $\boldsymbol{c}$. Thus

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \prod_{i=1}^{L} p(w_i|\boldsymbol{w}_{1:i-1}, \boldsymbol{Y}_{1:T}) \approx \prod_{t=1}^{T} p(w_i|w_{i-1}, \boldsymbol{h}_{i-2}, \boldsymbol{c}) \tag{1.80}$$

where $\boldsymbol{c} = \phi(\boldsymbol{Y}_{1:T})$. The power of this encoder-decoder framework compared to standard sequence kernel score-spaces is that the fixed length mapping is optimised in an end-to-end fashion with the recognition system.

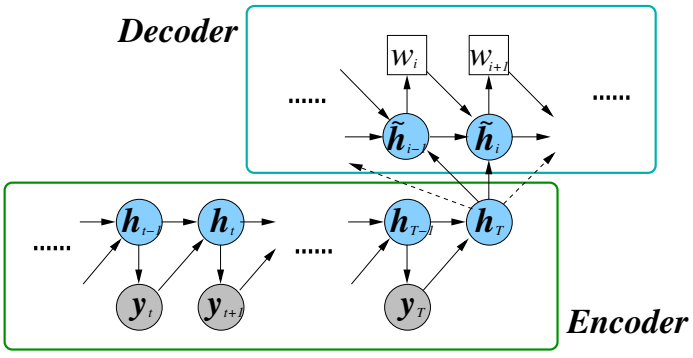


Figure 1.9 *RNN-based encode-decoder model*

The original encoder-decoder model was based on RNNs. This is illustrated in figure 1.9. Here the vector $\boldsymbol{c}$ is set to the final value of the history vector the encoder RNN,

$\boldsymbol{h}_T$. An RNN is also used as the decoder for the word-sequence. Thus

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) \approx \prod_{i=1}^{L} P(w_i|\boldsymbol{w}_{1:i-1}, \boldsymbol{h}_T) \approx \prod_{i=1}^{L} P(w_i|\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_T) \tag{1.81}$$

Though it is possible to use this form of model for speech recognition there is significant loss of information by encoding the complete observation sequence into a single vector. A more powerful form of model is based on

$$P(\boldsymbol{w}_{1:L}|\boldsymbol{Y}_{1:T}) = \prod_{i=1}^{L} p(w_i|\boldsymbol{w}_{1:i-1}, \boldsymbol{Y}_{1:T}) \approx \prod_{t=1}^{T} p(w_i|\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{c}_i) \tag{1.82}$$

where the complete set of feature history vectors $\boldsymbol{H}_{1:T} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T\}$ is used,

$$\boldsymbol{c}_i = \boldsymbol{\phi}(\boldsymbol{w}_{1:i-1}, \boldsymbol{Y}_{1:T}) = \boldsymbol{\phi}(\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{H}_{1:T}) \tag{1.83}$$

Attention-based approaches of this form have been used for speech recognition (Chorowski, Bahdanau, Serdyuk, Cho & Bengio 2015)

$$\boldsymbol{c}_i = \sum_{\tau=1}^{T} \alpha_{i\tau} \boldsymbol{h}_\tau; \quad \alpha_{i\tau} = \frac{\exp(e_{i\tau})}{\sum_{j=1}^{L} \exp(e_{ij})}, \quad e_{i\tau} = f^{\mathrm{e}}\left(\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_\tau\right) \tag{1.84}$$

One issue that has been observed with these forms of model is that the attention can drift in a non-causal (left-to-right) fashion.

1.6 Probability Distributions over Sequences

A general problem, applicable to other research areas, is how to obtain a distribution over a variable length sequence of observations. It is necessary to be able to generate samples from sequences whose length has never been seen in the training data. Distributions over the data have already been described in section 1.3. Take the example of an RNN generative model. Here

$$p(\boldsymbol{Y}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{Y}_{1:t-1}) \approx p(\boldsymbol{y}_1) \prod_{t=2}^{T} p(\boldsymbol{y}_t|\boldsymbol{h}_{t-1}) \tag{1.85}$$

where

$$\boldsymbol{h}_t = \boldsymbol{f}(\boldsymbol{y}_t, \boldsymbol{h}_{t-1}) \tag{1.86}$$

Generating an sequence from length T from this distribution simply requires generating an initial sample from $p(\boldsymbol{y}_1)$ and then recursively generating samples. Usually Gaussian distributions for continuous variables, and multinomial distributions for discrete variables, are used for th form of the distributions. Extensions of this form of model, bidirectional models and latent variable models, are also described in section 1.3.

One issue that can arise from simply relying on the history vector from the RNN, $\boldsymbol{h}_{t-1}$, to contain all the necessary information from the complete back-history of samples $\boldsymbol{Y}_{1:t-1}$. In practice this is not always possible. This section discusses the general issue

of distributions for variable length sequences, and the use of the product of experts framework (Hinton 1999) to address this problem.

1.6.1 Product of Experts

In the product of experts framework for sequence data the temporal structure of the data is modelled by a set of experts. The overall distribution of a T-length sequence can then be expressed in terms of the product of the information from all the experts. Thus

$$p(\boldsymbol{Y}_{1:T}) = \frac{1}{Z}\left(\prod_{e=1}^{E} p(\mathcal{F}^{(e)}(\boldsymbol{Y}_{1:T}))^{\alpha^{(e)}}\right) \tag{1.87}$$

or expressed as a log-probability

$$\log(p(\boldsymbol{Y}_{1:T})) = -\log(Z) + \sum_{e=1}^{E} \alpha^{(e)} \log(p(\mathcal{F}^{(e)}(\boldsymbol{Y}_{1:T}))) \tag{1.88}$$

Here $\mathcal{F}_e(\boldsymbol{Y}_{1:T})$ is the feature functions associated with an expert that yields a value from the observation sequence $\boldsymbol{Y}_{1:T}$. The challenge is to enable the distributions to be obtained for any value of T, the core issue in sequence modelling. As sequences are being dealt with there is additional flexibility in the nature of the experts compared to non-sequence, static, tasks. The experts can be split into two distinct groups

- fixed-length sub-sequence, $\mathcal{F}_{\mathtt{f}}^{(e)}(\boldsymbol{Y}_{t-N_e:t+N_e})$. This may include information about the current value, local gradient and acceleration;
- attributes of the complete sequence, $\mathcal{F}_{\mathtt{s}}^{(e)}(\boldsymbol{Y}_{1:T})$. The most common example of this is the use of global variance.

In both cases it is possible to model any arbitrary length sequence, though the approaches to handling variable length data is very different. Now [12]

$$\begin{aligned}&\log(p(\boldsymbol{Y}_{1:T})) = \\ &-T\log(Z) + \sum_{t=1}^{T}\sum_{e=1}^{E_{\mathtt{f}}} \alpha_{\mathtt{f}}^{(e)} \log(p(\mathcal{F}_{\mathtt{f}}^{(e)}(\boldsymbol{Y}_{t-N_e:t+N_e}))) + \sum_{e=1}^{E_{\mathtt{s}}} \alpha_{\mathtt{s}}^{(e)} \log(p(\mathcal{F}_{\mathtt{s}}^{(e)}(\boldsymbol{Y}_{1:T})))\end{aligned} \tag{1.89}$$

As with standard statistical modelling approaches training and inference needs to be considered. There are two approaches to training the models. First each of the experts is separately trained and only combined at the end to yield the final distribution. This simplifies the training problem as the normalisation term Z is not considered during training. Additionally at synthesis time where either a sample sequence, or the mean sequence is used it is not needed. The second approach is to optimise the likelihood of the set of experts generating the training data. These two approaches will be discussed in more detail in section 1.7.

[12] For simplicity of notation "end-effects" are being ignored here so that the summation is not a function of the maximum span of all the experts. The span is also specified as symmetric.

1.6.2 Gaussian Experts and Linear Feature Functions

The general product of experts system described above has not discussed the form of the expert distribution, or the nature of the expert. The simplest approach is to restrict the experts to all be Gaussian acting on a fixed-length sub-sequence, and the feature functions to be linear (Zen et al. 2012). In this situation it is possible to define a linear transformation $\mathbf{A}^{(e)}$ for each expert e, such that for a particular expert e

$$\mathcal{F}_{\mathtt{f}}^{(e)}(\boldsymbol{Y}_{t-N_e:t+N_e}) = \mathbf{a}^{(e)\mathsf{T}} \begin{bmatrix} \boldsymbol{y}_{t-N_e} \\ \vdots \\ \boldsymbol{y}_{t+N_e} \end{bmatrix} = \mathbf{a}^{(e)\mathsf{T}} \boldsymbol{y}_t^{(e)} \sim \mathcal{N}(\mu^{(e)}, \sigma^{(e)2}) \tag{1.90}$$

Using this form of representation it is possible to define for a T-length sequence a mean vector, $\overline{\boldsymbol{\mu}}_T$, covariance matrix, $\overline{\boldsymbol{\Sigma}}_T$, and transformation matrix, $\overline{\mathbf{A}}_T$. For example if $N_e = 1$ (so $\boldsymbol{y}_t^{(1)} = \boldsymbol{y}_t^{(2)} = \boldsymbol{y}_t^{(e)}$) [13] for two experts and three, d-dimensional, observations (with zero padding) , $\mathbf{0}^d$ indicates a d-length *row* vector of zeros,

$$\begin{bmatrix} \mathbf{a}^{(1)\mathsf{T}} & \mathbf{0}^{3d} & \mathbf{0}^{3d} \\ \mathbf{a}^{(2)\mathsf{T}} & \mathbf{0}^{3d} & \mathbf{0}^{3d} \\ \mathbf{0}^{3d} & \mathbf{a}^{(1)\mathsf{T}} & \mathbf{0}^{3d} \\ \mathbf{0}^{3d} & \mathbf{a}^{(2)\mathsf{T}} & \mathbf{0}^{3d} \\ \mathbf{0}^{3d} & \mathbf{0}^{3d} & \mathbf{a}^{(1)\mathsf{T}} \\ \mathbf{0}^{3d} & \mathbf{0}^{3d} & \mathbf{a}^{(2)\mathsf{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1^{(e)} \\ \boldsymbol{y}_2^{(e)} \\ \boldsymbol{y}_3^{(e)} \end{bmatrix} = \begin{bmatrix} \mathbf{a}^{(1)\mathsf{T}} & \mathbf{0}^{d} & \mathbf{0}^{d} \\ \mathbf{a}^{(2)\mathsf{T}} & \mathbf{0}^{d} & \mathbf{0}^{d} \\ \mathbf{0}^{d} & \mathbf{a}^{(1)\mathsf{T}} & \mathbf{0}^{d} \\ \mathbf{0}^{d} & \mathbf{a}^{(2)\mathsf{T}} & \mathbf{0}^{d} \\ \mathbf{0}^{d} & \mathbf{0}^{d} & \mathbf{a}^{(1)\mathsf{T}} \\ \mathbf{0}^{d} & \mathbf{0}^{d} & \mathbf{a}^{(2)\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \mathbf{0} \end{bmatrix} = \overline{\mathbf{A}}_3 \begin{bmatrix} \mathbf{0} \\ \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \mathbf{0} \end{bmatrix} \tag{1.91}$$

The distribution for this transformed data can be expressed as

$$\overline{\mathbf{A}}_3 \begin{bmatrix} \mathbf{0} \\ \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \mathbf{0} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu^{(1)} \\ \mu^{(2)} \\ \mu^{(1)} \\ \mu^{(2)} \\ \mu^{(1)} \\ \mu^{(2)} \end{bmatrix}, \begin{bmatrix} \sigma^{(1)2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^{(2)2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^{(1)2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^{(2)2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^{(1)2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^{(2)2} \end{bmatrix} \right) = \mathcal{N}\left(\overline{\boldsymbol{\mu}}_3, \overline{\boldsymbol{\Sigma}}_3\right) \tag{1.92}$$

As the overall "transformed" sequence distribution is Gaussian, and the relationship of the original sequence to the transformed sequence is linear, it can be shown that the resulting distribution of the original sequence is Gaussian. This distribution can be expressed in the following form for a general T length sequence

$$p(\boldsymbol{Y}_{1:T}) = \mathcal{N}\left(\boldsymbol{Y}_{1:T}; \left(\overline{\mathbf{A}}_T^{\mathsf{T}} \overline{\boldsymbol{\Sigma}}_T^{-1} \overline{\mathbf{A}}_T\right) \overline{\mathbf{A}}_T^{\mathsf{T}} \overline{\boldsymbol{\mu}}_T, \left(\overline{\mathbf{A}}_T^{\mathsf{T}} \overline{\boldsymbol{\Sigma}}_T^{-1} \overline{\mathbf{A}}_T\right)^{-1}\right) \tag{1.93}$$

Thus the effective mean, $\tilde{\boldsymbol{\mu}}_T$, and covariance matrix, $\tilde{\boldsymbol{\Sigma}}_T$ of the complete T-length observation sequence may be expressed as

$$\tilde{\boldsymbol{\mu}}_T = \tilde{\boldsymbol{\Sigma}}_T \overline{\mathbf{A}}_T^{\mathsf{T}} \overline{\boldsymbol{\mu}}_T, \qquad \tilde{\boldsymbol{\Sigma}}_T = \left(\overline{\mathbf{A}}_T \overline{\boldsymbol{\Sigma}}_T \overline{\mathbf{A}}_T^{\mathsf{T}}\right)^{-1} \tag{1.94}$$

So the distribution depends on both the parameters of the experts and the length of the observation sequence. For long sequences this inversion process can become computationally expensive.

It is sometimes useful to relax the restriction of only extracting linear transformations

[13] This is not a restriction as it is possible to specify N_e to be the maximum sub-sequence length.

of sub-sequences for the experts. One example often used for speech synthesis is the global variance expert (Toda & Tokuda 2007). Here the variance of each dimension, i, of the observation sequence is extracted as a feature

$$\mathcal{F}_{\texttt{gv}i}(\boldsymbol{Y}_{1:T}) = \sigma^2_{\texttt{gv}i} = \sum_{t=1}^{T} \left(y_{ti} - \mu_{\texttt{gv}i}\right)^2 \tag{1.95}$$

where $\mu_{\texttt{gv}i}$ is the mean of dimension i of the observation sequence. It may also be useful not to limit the form of the distribution to be Gaussian. In these cases it is not trivial to optimise the model parameters. To address this contrastive divergence based approached can be used (Zen et al. 2012) to train the model parameters.

1.7 Training Criteria and Loss Functions

One of the interesting aspects of sequence-to-sequence models is the nature of the training criterion. Here, only supervised training will be discussed. Extensions to lightly-supervised, semi-supervised and unsupervised are also possible (Lamel, Gauvain & Adda 2002). Only give a top-level description of the criterion will be give. The details of the optimisation approaches, and issues will be discussed elsewhere. The general expression for training is

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \{\mathcal{F}(\boldsymbol{\lambda}; \mathcal{D})\} \tag{1.96}$$

where for n utterances of supervised data

$$\mathcal{D} = \left\{\left(\boldsymbol{Y}^{(1)}_{1:T^{(1)}}, \boldsymbol{w}^{(1)}_{1:L^{(1)}}\right), \ldots, \left(\boldsymbol{Y}^{(n)}_{1:T^{(n)}}, \boldsymbol{w}^{(n)}_{1:L^{(n)}}\right)\right\} \tag{1.97}$$

The *de facto* standard approach for training systems is to use maximum likelihood (ML). However this term has different meanings depending on where generative or discriminative models are being used:

- **generative model**: here the model parameters are tuned to maximise the joint probability

$$\mathcal{F}_{\texttt{jnt}}(\boldsymbol{\lambda}; \mathcal{D}) = \sum_{i=1}^{n} \log(p(\boldsymbol{w}^{(i)}_{1:L^{(i)}}, \boldsymbol{Y}^{(i)}_{1:T^{(i)}}; \boldsymbol{\lambda})) \tag{1.98}$$

  More commonly the acoustic model is trained separately from the language model. Thus

$$\mathcal{F}_{\texttt{ml}}(\boldsymbol{\lambda}; \mathcal{D}) = \sum_{i=1}^{n} \log(p(\boldsymbol{Y}^{(i)}_{1:T^{(i)}} | \boldsymbol{w}^{(i)}_{1:L^{(i)}}; \boldsymbol{\lambda})) \tag{1.99}$$

- **discriminative model**: here the model parameters are optimised to maximise the probability of the word-sequence. To distinguish this form of optimisation from the generative, this is often referred to as the *Conditional Maximum Likelihood* (CML) training criterion (in speech recognition also referred to as Maximum Mutual Information (MMI) training (Bahl, Brown, de Souza & Mercer 1986)).

$$\mathcal{F}_{\texttt{cml}}(\boldsymbol{\lambda};\mathcal{D}) = \sum_{i=1}^{n} \log(p(\boldsymbol{w}^{(i)}_{1:L^{(i)}}|\boldsymbol{Y}^{(i)}_{1:T^{(i)}};\boldsymbol{\lambda})) \tag{1.100}$$

1.7.1 Speech Recognition Criteria

Speech recognition is a classification task, usually based on Bayes' decision rule (repeating equation 1.8 for clarity)

$$\hat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \{P(\boldsymbol{w}|\boldsymbol{Y}_{1:T};\boldsymbol{\lambda})\} \tag{1.101}$$

Discriminative models directly estimate the word posterior, $P(\boldsymbol{w}|\boldsymbol{Y}_{1:T};\boldsymbol{\lambda})$. For generative models this posterior is obtained from

$$P(\boldsymbol{w}|\boldsymbol{Y}_{1:T};\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{Y}_{1:T})} p(\boldsymbol{Y}_{1:T}|\boldsymbol{w};\boldsymbol{\lambda})P(\boldsymbol{w};\boldsymbol{\lambda}) \tag{1.102}$$

where Z is the normalisation term obtained by summing the joint distribution over all possible word sequences. Thus during recognition time both generative and discriminative models require the calculation of the word posterior.

Initially generative models were trained using maximum likelihood for both acoustic and language models. Later these generative models were trained using CML applying Bayes' rule to convert the joint distribution to a word posterior. For ASR other forms of discriminative criteria have also been examined. Note the sane criteria can be used for generative and discriminative models as both can represent the word posterior.

CML training is well motivated if the aim of the process is to obtain the system that is linked to sentence (word sequence) level performance. Given that speech processing task need to generate sub-sentence, and sub-word, models to handle the vast number of possible sentences. One alternative, popular approach in speech recognition, is to examine Minimum Bayes' Risk (MBR) training (Kaiser, Horvat & Kačič 2000, Povey & Woodland 2002, Byrne 2006, Gibson & Hain 2006). Here the models parameters are estimated by *minimising*

$$\mathcal{F}_{\texttt{mbr}}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{Y}^{(i)}_{1:T^{(i)}};\boldsymbol{\lambda})\mathcal{L}(\boldsymbol{w},\boldsymbol{w}^{(i)}_{1:L^{(i)}}) \tag{1.103}$$

where $\mathcal{L}(\boldsymbol{w},\boldsymbol{w}^{(i)}_{1:L^{(i)}})$ is the *loss* between the reference word sequence, $\boldsymbol{w}^{(i)}_{1:L^{(i)}}$, and the "hypothesis". The loss function can be measured at various levels. Common forms are, sentence, word, phone and state. This form of criterion, and related criteria such as Minimum Classification Error (MCE) (Juang & Katagiri 1992), have been applied to a range of speech recognition tasks.

An interesting modification to the standard forms of MBR training in speech recognition, is to consider the loss function at the frame level , referred to here as *minimum frame risk* (Zheng & Stolcke 2005). Here

$$\mathcal{F}_{\texttt{mfr}}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{\boldsymbol{w}} \sum_{\boldsymbol{\theta}:|\boldsymbol{\theta}|=T^{(i)}} P(\boldsymbol{w}|\boldsymbol{\theta};\boldsymbol{\lambda})P(\boldsymbol{\theta}|\boldsymbol{Y}^{(i)}_{1:T^{(i)}};\boldsymbol{\lambda})\mathcal{L}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}^{(i)}_{1:T^{(i)}}) \tag{1.104}$$

where $\hat{\boldsymbol{\theta}}^{(i)}_{1:T^{(i)}}$ is the reference frame-level alignment. The loss at the frame can then be considered at various levels in the same fashion as standard Bayes' risk training.

The final class of criteria introduce a *margin* into the estimation process. One form that has been used for discriminative models is to *minimise* the following large-margin criterion (Sha & Saul 2007, Zhang & Gales 2012)

$$\mathcal{F}_{\texttt{lm}}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \max_{\boldsymbol{w} \neq \boldsymbol{w}^{(i)}_{1:L^{(i)}}} \left[\mathcal{L}(\boldsymbol{w}, \boldsymbol{w}^{(i)}_{1:L^{(i)}}) - \log \left(\frac{P(\boldsymbol{w}^{(i)}_{1:L^{(i)}} | \boldsymbol{Y}^{(i)}_{1:T^{(i)}}; \boldsymbol{\lambda})}{P(\boldsymbol{w} | \boldsymbol{Y}^{(i)}_{1:T^{(i)}}; \boldsymbol{\lambda})} \right) \right] \tag{1.105}$$

One of the interesting aspects of this criterion is that the normalisation term, that usually impacts the training, is not part of the criterion. The posterior ratio means that it is cancelled from numerator and denominator. When a log-linear model used, and only the best state or phone sequence taken into account, this yields a Structured Support Vector Machine (SSVM) (Tsochantaridis, Joachims, Hofmann & Altun 2005).

1.7.2 Speech Synthesis Criteria

Speech synthesis is a regression process, generative in nature. Repeating equation 1.12

$$\hat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} \{ p(\boldsymbol{Y} | \boldsymbol{w}_{1:L}; \boldsymbol{\lambda}) \} \tag{1.106}$$

This ties in closely with the standard ML criterion for generative models. If multiple experts are used to generate smooth trajectories, then the following criterion is maximised

$$\mathcal{F}_{\texttt{ml}}(\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{\lambda}} \left\{ \sum_{i=1}^{n} \log \left(p \left(\begin{bmatrix} \mathcal{F}_1(\boldsymbol{Y}^{(i)}_{1:T^{(i)}} | \boldsymbol{w}^{(i)}_{1:L^{(i)}}) \\ \vdots \\ \mathcal{F}_E(\boldsymbol{Y}^{(i)}_{1:T^{(i)}} | \boldsymbol{w}^{(i)}_{1:L^{(i)}}) \end{bmatrix}; \boldsymbol{\lambda} \right) \right) \right\} \tag{1.107}$$

where $\boldsymbol{\lambda}$ comprises the parameters for the complete set of E experts. This maximises the likelihood of each of the individual experts rather than the generated trajectories.

An alternative approach is based on the trajectories generated by the experts. The first option, similar to ML training, is to find the parameters of the experts that maximise the likelihood of complete trajectory (Zen et al. 2007).

$$\mathcal{F}_{\texttt{traj}}(\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{\lambda}} \left\{ \sum_{i=1}^{n} \log \left(p \left(\boldsymbol{Y}^{(i)}_{1:T^{(i)}} | \boldsymbol{w}^{(i)}_{1:L^{(i)}}; \boldsymbol{\lambda} \right) \right) \right\} \tag{1.108}$$

This is the criterion related to the trajectory model, Figure 1.4. A second criterion is based on the "difference" between the trajectory generated by the model and the training data trajectory (Wu & Wang 2006). Here

$$\mathcal{F}_{\texttt{mge}}(\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{\lambda}} \left\{ \sum_{i=1}^{n} \sum_{t=1}^{T^{(i)}} \mathcal{D}(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t) \right\} \tag{1.109}$$

where $\mathcal{D}(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)$ is a frame-level distance, generation error, between the observation at time instance t, $\boldsymbol{y}_t$, and the ML-generated trajectory $\hat{\boldsymbol{y}}_t$. When the frame-level distance measure is euclidean then this is related to the trajectory model, with the constraint

that the variance of the trajectory is an identity matrix. However this form of criterion has significant flexibility in terms of the distance being used, allowing for example perceptual based weighting to be included.

One important difference between the trajectory criterion and the generation error is in the handling of the alignment, duration, model $p(\boldsymbol{\theta}_{1:T}|\boldsymbol{w}_{1:L})$. For the MGE criterion "reference" durations need to be used as the sequences need to be time-aligned [14]. The alignment model training is not integrated with the expert parameter training. Conversely for the trajectory criterion an integrated approach can be adopted.

1.8 Summary

This report has briefly reviewed sequence modelling in general and more specifically sequence-to-sequence modelling. Both generative and discriminative forms of sequence-to-sequence models are described. Central to many of these are conditional independence assumptions that allow variable length sequences to be efficiently modelled. Additionally other forms of model, incorporating deep-learning approaches are described. Finally forms of distribution over sequences are described, as well as various forms of speech recognition and synthesis criteria.

[14] More generally sequence kernels could be used.

Notes

Bibliography

Bahl, L. R., Brown, P. F., de Souza, P. V. & Mercer, R. L. (1986), Maximum mutual information estimation of hidden markov model parameters for speech recognition, *in* 'Proceedings of IEEE international conference on acoustics, speech, and signal processing', Tokyo, Japan, pp. 49–52.

Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C. (2003), 'A neural probabilistic language model', *journal of machine learning research* **3**(Feb), 1137–1155.

Bisani, M. & Ney, H. (2008), 'Joint–sequence models for grapheme–to–phoneme conversion', *Speech Comm.* **50**(5), 434–451.

Bishop, C. (1994), Mixture density networks, *in* 'Tech. Rep. NCRG/94/004, Neural Computing Research Group, Aston University'.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer Verlag.

Bourlard, H. & Morgan, N. (1994), *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers.

Byrne, W. J. (2006), 'Minimum bayes risk estimation and decoding in large vocabulary continuous speech recognition', *IEICE Transactions on Information and Systems: Special Issue on Statistical Modelling for Speech Recognition* **E89-D**(3), 900–907.

Chen, X., Liu, X., Wang, Y., Gales, M. J. & Woodland, P. C. (2016), 'Efficient training and evaluation of recurrent neural network language models for automatic speech recognition', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**(11), 2146–2157.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K. & Bengio, Y. (2015), Attention-based models for speech recognition, *in* 'Advances in Neural Information Processing Systems', pp. 577–585.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014), 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *arXiv* .

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C. & Bengio, Y. (2015), 'A recurrent latent variable model for sequential data', *arXiv* **abs/1506.02216**.

Cortes, C., Haffner, P. & Mohri, M. (2004), 'Rational kernels: Theory and algorithms', *Journal of Machine Learning Research* **5**, 1035–1062.

Gales, M. J. F., Watanabe, S. & Fosler-Lussier, E. (2012), 'Structured discriminative models for speech recognition: An overview', *IEEE Signal Process. Mag.* **29**(6), 70–81.

Gales, M. J. F. & Young, S. J. (2007), 'The application of hidden Markov models in speech recognition', *Foundations and Trends in Signal Processing* **1**(3), 195–304.

Gibson, M. & Hain, T. (2006), Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition., *in* 'Interspeech', Vol. 6, pp. 2406–2409.

Graves, A. (2012), 'Sequence transduction with recurrent neural networks', *arXiv preprint arXiv:1211.3711* .

Graves, A., Fernández, S., Gomez, F. & Schmidhuber, J. (2006), Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, *in* 'Proc. ICML', ACM, pp. 369–376.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. & Kingsbury, B. (2012), 'Deep neural networks for acoustic modeling in speech recognition', *IEEE Signal Processing Magazine* **29**(6), 82–97.

Hinton, G. E. (1999), Products of experts, *in* 'Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN 99)', pp. 1–6.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural Comput.* **9**(8), 1735–1780.

Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R. & Ney, H. (2016), 'LSTM, GRU, highway and a bit of attention: an empirical overview for language modeling in speech recognition', *Proc. Interspeech* .

Juang, B.-H. & Katagiri, S. (1992), 'Discriminative learning for minimum error classification', *IEEE Transactions on Signal Processing* .

Kaiser, J., Horvat, B. & Kačič, Z. (2000), A novel loss function for the overall risk criterion based discriminative training of hmm models, *in* 'Proceedings of the sixth international conference on spoken language processing', Vol. 2, Beijing, China, pp. 887–890.

Kuo, H.-K. & Gao, Y. (2006), 'Maximum entropy direct models for speech recognition', *IEEE/ACM Trans. Audio, Speech, Language Process.* .

Lamel, L., Gauvain, J.-L. & Adda, G. (2002), 'Lightly supervised and unsupervised acoustic model training', *Computer Speech & Language* **16**(1), 115–129.

Lauritzen, S. L. (1996), *Graphical models*, Vol. 17, Clarendon Press.

Liu, X., Chen, X., Wang, Y., Gales, M. J. & Woodland, P. C. (2016), 'Two efficient lattice rescoring methods using recurrent neural network language models', *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* **24**(8), 1438–1449.

Lu, L., Kong, L., Dyer, C., Smith, N. A. & Renals, S. (2016), 'Segmental recurrent neural networks for end-to-end speech recognition', *arXiv preprint arXiv:1603.00223* .

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J. & Khudanpur, S. (2010), Recurrent neural network based language model., *in* 'Interspeech', Vol. 2, p. 3.

Povey, D. & Woodland, P. C. (2002), Minimum phone error and I-smoothing for improved discriminative training, *in* 'Proceedings of IEEE international conference on acoustics, speech, and signal processing', pp. I–105–I–108.

Robinson, T. & Fallside, F. (1991), 'A recurrent error propagation network speech recognition system', *Computer Speech & Language* **5**(3), 259–274.

Rosti, A.-V. I. & Gales, M. J. (2001), 'Generalised linear Gaussian models'.

Roweiss, S. & Ghahramani, Z. (1999), 'A unifying view of linear Gaussian models', *Neural Comput.* **11**, 305–345.

Schuster, M. & Paliwal, K. K. (1997), 'Bidirectional recurrent neural networks', *IEEE Trans. Signal Process.* **45**(11), 2673–2681.

Sha, F. & Saul, L. K. (2007), Large margin hidden markov models for automatic speech recognition, *in* B. Schölkopf, J. Platt & T. Hofmann, eds, 'Advances in Neural Information Processing Systems 19', MIT Press, pp. 1249–1256.

Smith, N. & Gales, M. (2002), 'Speech recognition using svms', *Advances in neural information processing systems* **2**, 1197–1204.

Sutskever, I., Vinyals, O. & Le, Q. V. (2014), Sequence to sequence learning with neural networks, *in* 'Advances in neural information processing systems', pp. 3104–3112.

Toda, T. & Tokuda, K. (2007), 'A speech parameter generation algorithm considering global variance for HMM-based speech synthesis', *IEICE T. Inf. Syst.* **90**(5), 816–824.

Tsochantaridis, I., Joachims, T., Hofmann, T. & Altun, Y. (2005), 'Large margin methods for structured and interdependent output variables', *Journal of machine learning research* **6**(Sep), 1453–1484.

Viterbi, A. (1967), 'Error bounds for convolutional codes and an asymptotically optimum decoding algorithm', *IEEE Trans. Inf. Theory* **13**(2), 260–269.

Wu, Y.-J. & Wang, R.-H. (2006), Minimum generation error training for HMM-based speech synthesis, *in* 'Proc. ICASSP', Toulouse.

Zen, H., Gales, M. J., Nankaku, Y. & Tokuda, K. (2012), 'Product of experts for statistical parametric speech synthesis', *IEEE/ACM Trans. Audio, Speech, Language Process.* **20**(3), 794–805.

Zen, H. & Sak, H. (2015), Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis, *in* 'Proc. ICASSP', Vol. 40, Brisbane, Australia, pp. 4470–4474.

Zen, H. & Senior, A. (2014), Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis, *in* 'Proc. ICASSP', Vol. 39, Florence, Italy, pp. 3844–3848.

Zen, H., Tokuda, K. & Kitamura, T. (2007), 'Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences', *Comput. Speech Lang.* **21**(1), 153–173.

Zhang, S.-X. & Gales, M. J. F. (2012), 'Structured SVMs for automatic speech recognition', *IEEE Transactions on Audio, Speech, and Language Processing* .

Zheng, J. & Stolcke, A. (2005), Improved discriminative training using phone lattices, *in* 'Proceedings of the ninth European conference on speech communication and technology', Lisbon, Portugal, pp. 2125–2128.