CHAPTER 3

Theory

This section will provide an overview of aspects of the theory of the models and signal processing techniques used in this dissertation. This will include an explanation of, and an introduction to:

Hidden Markov models and how they are used for recognition

Signal processing, feature extraction and selection

3.1 Hidden Markov models

Physical processes generally produce observable outputs that can be represented by signal models. These models allow us to learn a great deal about the process, without having the actual signal source around. There are several choices for a user when it comes to the types of signal model that can be used to characterise the properties of the signal of interest. According to Rabiner (1989), signal models can broadly be divided into two groups:

- Deterministic models which exploit the known properties of the signal (e.g. the signal is a sine wave or a sum of exponentials).
- Statistical models where one tries to characterise only the statistical properties of the process. (e.g. Gauss processes, Poisson processes, Markov processes).

Under the statistical model it is assumed that the process can be described by a parametric random process for which the parameters can be estimated by means of a well defined formulation.

These signals can further be divided into discrete and continuous signals. Statistical models can also be stationary (statistical properties are time invariant) or non-stationary (statistical properties vary with time). Hidden Markov models (or Markov source in older literature) falls into the category of non-stationary statistical models.

3.1.1 Defining the HMM

A hidden Markov model can be defined as finite state machine that functions in discrete time. Each state in the HMM contains the definition of some stochastic process (i.e. a Probability Density Function (PDF) or an AR-model). At each time step the HMM emits an observation from one of its states. A signal/observation sequence may then be produced by taking a random walk (defined by a Markov process) "within" the states. This random walk is dependant on the transition probabilities. To clarify this consider figure 3.1 which shows a network diagram of a 3-state HMM. The lines connecting the states (numbered $1 - 3$) represent state transition probabilities. The state transition probabilities are the probabilities that the HMM, currently in state i will transit to state j for the next time step. An HMM can also stay in its current state for the next time step. This is shown as little "loopbacks" on the figure. The HMM is therefore a doubly stochastic process in the fact that it is a random process for which the variables are determined by a random Markov process. The HMM is also in actual fact, a statistical signal generator although it is not used as a signal source. It is rather used as a vehicle for probabilistic inference. This will be explained later on in this chapter.

The reason for its name is that, during training the state sequence cannot be observed from the training sequences. The state sequence is therefore "hidden", hence the name. The training goal is therefore to infer the state sequence and to determine the state process parameters from the training sequences. It should be noted here that training sequences are the same as the observation sequences. Once the state transition probabilities and the state process parameters are determined the model can be used for classification. The technique for classification used in this study is called "scoring" and is described on page 13.

The emissions from the states can be of a continuous or a discrete nature. Discrete emissions are usually symbols while continuous emissions may be a real valued numbers within a certain range.

Definitions

The HMM used for this project will have discrete emissions and discrete states. This is a very specific subclass of HMMs and the interested reader should consult Elliott et al. (1995) for a more advanced and general description of HMMs [1] [2]. The notation used

[1] Some additional theory on HMMs can be found in appendix A

[2] These definitions are from Narada Warakagoda´s website at http://jedlik.phy.bme.hu/ ˜gerjanos/HMM/node3.htm.

117512244
b16427014

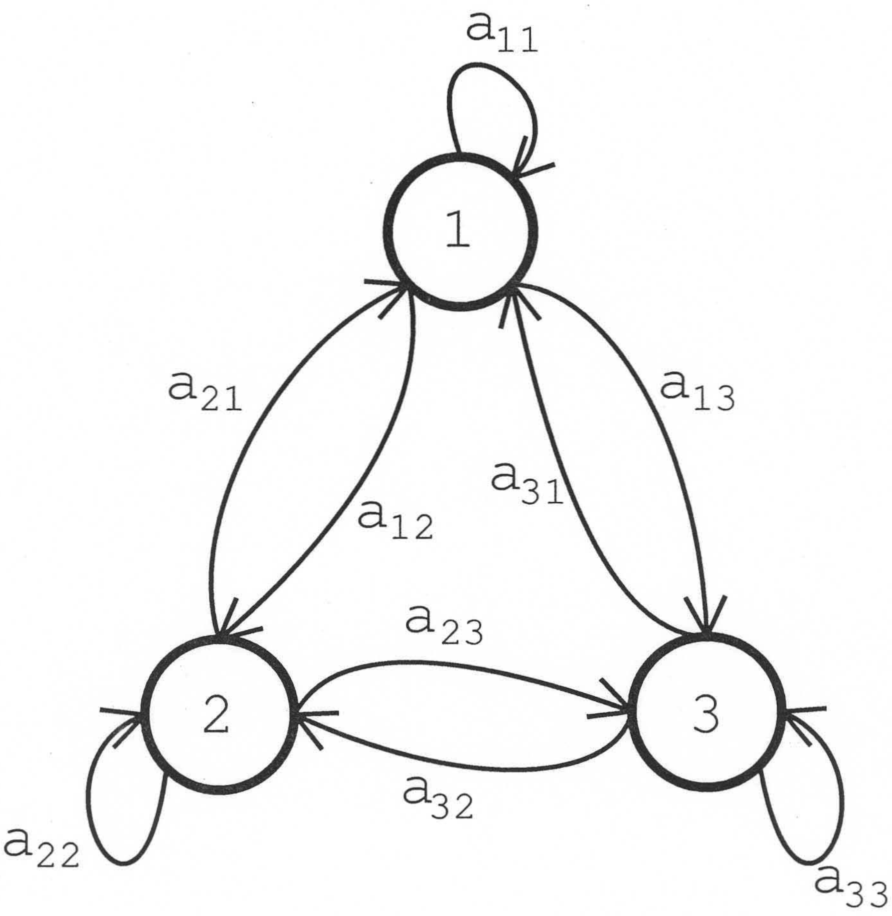

Figure 3.1: A directed state-transition graph of an ergodic 3-state HMM

throughout this text will be that of Rabiner (1989)[3]

An HMM is completely defined by the following parameters:

- State transition matrix, A. This defines the probability that the model, currently in state i will transit to state j for the next time step. This will be written as a_{ij}. The reader is again referred to figure 3.1. A will thus always be square and have the form:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{3.1}$$

  The number of states that the model can then assume, N is equal to the number of columns in A. A is also subjected to the normal stochastic constraints namely:

$$a_{ij} \geq 0 \qquad \text{with } 0 \leq i, j \leq N$$

  and

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \text{for all } i$$

- Probability distribution for each state. This probability distribution will be denoted with B and is defined as follows: b_{ik} is the probability that the model, currently at state i will emit the k-th symbol in the defined alphabet of discrete emissions. As was previously mentioned the HMM will have discrete emissions defined within an alphabet with total number of M symbols. For a HMM with i states and M symbols, B will then have the form:

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{iM} \end{pmatrix} \tag{3.2}$$

  As with A, B is also subjected to the normal stochastic constraints:

$$b_{ik} \geq 0 \qquad \text{with } 0 \leq i, k \leq M$$

  and

$$\sum_{k=1}^{M} b_{ik} = 1 \qquad \text{for all } i$$

- Initial state probability distribution named π. π is the probability distribution that describes the likelihood that a HMM will start in state i. The normal stochastic constraints apply.

[3] This is also a very good starting place for readers who are new to the subject of HMMs.

Once A, B and π are defined, one has a complete HMM. To shorten the notation a specific model will be denoted as λ. $\lambda = f(A, B, \pi)$, viz. Given λ, the HMM can be used to generate a sequence of observations,

$$O = \{o_1, o_2, o_3, \ldots, o_T\}$$

O is a vector that contains the emitted observations from time, $t = 1$ to time $t = T$, with T being the length of the sequence in discrete-time. When an HMM is to be trained, the signals need to be segmented into these observation sequences.

3.1.2 The three problems of HMMs

Once one has defined a HMM, λ, there are certain things that one usually wants to be able to do with it. In HMM literature one will read of the three problems of HMMs, which describe what the HMM will be used for. These are discussed in Rabiner (1989) in the form of 3 problems. These are:

1. Given a HMM model, λ and an observation sequence, $O = \{o_1, o_2, o_3, \ldots, o_n\}$, how is the probability, $P(O|\lambda)$ efficiently calculated? ($P(O|\lambda)$ is the probability that the HMM, λ produces the emission sequence, O.)

2. Given a HMM model, λ and an observation sequence, $O = \{o_1, o_2, o_3, \ldots, o_n\}$, how is the state sequence, that in some way optimally describes the observation sequence, chosen?

3. How can the model parameters A,B and π be chosen so as to maximise $P(O|\lambda)$?

The solution to problem 1 is used in this dissertation to score HMMs. Consider the scenario where one has different competing models that describe an observation set. The solution to problem 1 can then be used to select the model with the highest probability of producing the observation set in question.

The solution to problem 2, called the Viterbi algorithm is not used in this dissertation and thus falls outside of the scope of discussion. The reader is referred to Rabiner (1989) and Bengio (1999) for an in-depth description of this procedure.

Problem 3 does not have a known analytical solution to choose the model that maximises $P(O|\lambda)$. This makes it the most difficult problem of the HMMs.

The Forward Procedure

It was mentioned previously that classification can be done with HMMs with a technique called scoring. This is a procedure where the probability is calculated that a given HMM, say λ_1, will emit a certain sequence. In order to do this one needs to calculate the emission probabilities for the given sequence, for each possible state sequence. This quickly becomes intractable. Fortunately there exists an efficient recursive algorithm to do this. This algorithm is called the forward procedure and is discussed in Rabiner (1989). The result of the Forward procedure is called the forward probability and is denoted with an α.

For an HMM with N states and a sequence length of T the procedure works as follows:

1. Initialisation:

$$\alpha_1(i) = \pi_i b_i(O_1) \tag{3.3}$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \quad \text{and} \quad 1 \leq j \leq N \tag{3.4}$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.5}$$

There is no analytical solution to show what number of states will produce the best HMM for a specific application. It can however be said that a model with more states may perform better. This is because the amount of states in the HMM is directly related to its ability to model signal non-stationarities. More states unfortunately require more training data which may be difficult to come by.

Another problem encountered with the calculation of probabilities using HMMs is that of underflow. The numbers tend to be extremely small, well under machine precision for most computers. For this reason the probabilities are scaled and use is made of logarithmic probabilities. As the name implies, the logarithm of the probabilities are calculated and used in the algorithms. The properties of the probabilities are now slightly different. Whereas in the normal case where probabilities lie between 0 and 1, logarithmic probabilities lie between $-\infty$ and 0. With HMMs it is usually not strange to work with probabilities in the range of -100, which is a very small number indeed!

Training the hidden Markov model

This is the most difficult problem of the HMM. According to Rabiner (1989) there is no known way to analytically solve for the model parameters that maximises the probability of the observation sequence. The most common technique usually employed is the Baum-Welch method which, locally maximises λ for $P(O|\lambda)$. This method is equivalent the Expectation-Modification (EM) algorithm, which is a maximum likelihood approach.

There also exists some gradient based methods but usually the EM algorithm is preferred for its fast convergence properties. The EM technique also guaranties a finite improvement on each iteration. Conditions can be formulated so that gradient based method can be applied to the HMM and this is presented by Rabiner (1989). Kwon and Kim (1999) have devised a method that uses the EM algorithm together with a genetic algorithm to train the HMM and to a select a state topology. Good results are achieved but training is slow.

As with neural networks, HMMs also have an architecture that needs to decided on, eg. the number of states and the state topology. Faced with this problem Bicego et al. (2003) presented a strategy to sequentially prune the number of states in an HMM.

It is important to know what is being done when one trains an HMM. Training implies that the parameters that define the HMM are updated. As mentioned previously these parameters are:

- the state transition matrix, A
- the emission probability density function for each state B
- the initial state distribution, π

To do this the α-parameter is once again used. Three other similar variables are also introduced in order to make training possible. It is because of these three other quantities that the training algorithm will not be shown here. A thorough description can be found in Rabiner (1989). Alternatively there is also a shorter version in appendix A.

3.2 Signal processing

In order for any intelligent system to be applied to the data, the data first needed to be processed into a different form that would be usable by the system. There are some similarities between speech data and vibration data and the signals processing techniques used on them. Bunks et al. (2000) compares speech data to acceleration data from a helicopter gearbox. This can unfortunately not be used directly because machining data is fundamentally different from acceleration data form gearboxes. An altered version will be presented.

Data from machining processes and speech are both quasi-stationary. The speech data however stays stationary over intervals of approximately $10ms$, according to Bunks et al. (2000). Cutting processes may be of one of two types. Interrupted cutting, in which the cutting tool is in contact with the workpiece for only a fraction of each revolution, produce signals which are stationary for intervals of milliseconds. Continuous cutting, where the tool is in contact with the workpiece for the whole period of each revolution, on the other hand produce cutting signals which are stationary for longer periods of time.

Another difference is that speech data is recorded in relatively "quiet" environments. Vibration data from working environments may, in the very worst cases, have a signal to noise ratio, orders of magnitude lower than that of speech data.

Another is difficulty is that changes in tool condition produce only slight changes in the response of the tool holder which are recorded. This necessitates the use of signal features which compress the information content of the signal. This is also why, in this study, features for NN studies will be investigated for the use of HMM applications. Therefore owing to the different nature of the data from speech signals, the raw signal was not used. The data had to undergo a number of preprocessing steps. These were:

- *Segmentation* of the raw signals into intervals for which the features are calculated.
- *Detrending*, which removes the most dominant linear trend from the data. This is usually done for FFT analysis. After this the observation sequences have a mean of zero.
- *Feature extraction* whereby the salient features of the signals are extracted.
- *Feature Selection*, is applied so that only the features with the most information with respect to tool wear is used.
- *Feature space reduction* which condenses the selected features into the final product which was a 1-dimensional feature vector.
- *Discretisation and Construction* of observation sequences. The signals are firstly discretized into a number of levels then consecutive samples from the feature space are constructed into rows of observation sequences of a specific length.

3.2.1 Feature extraction

In order to learn most about tool wear, certain features are extracted from the data. Each feature has a characteristic behaviour that can be followed over time to reveal information about the health of the tool. It is in this way that features will be used in this study.

The extraction of features also compresses the data into a form, which can be handled with much more ease and efficiency. This is important for real-time implementation, which is the longterm goal for any project that hopes to see an industrial application.

Two types of features were investigated, time domain and frequency domain. These two will be discussed in the sections.

Features in the time domain

Features in time are usually figures that one would normally find in most statistical analyses. As a tool wears, in the case of flank wear, the wear land increases. The interaction surface of the workpiece and tool is then changed. This also alters the interaction of frictional forces between the two elements in the system. The result of this are changes in the dynamic characteristics of the system.

The features of the time domain are usually of a statistical nature. These features are also very fast to calculate which makes them very attractive for on-line applications. The interested reader may also review the implications of some of the statistical parameters used in a text such as Miller and Miller (1999).

The features that were investigated were:

- *Variance*, which is the second statistical moment of the data. Because of detrending the mean of the data is 0 which makes the variance of the data equal to the square of the RMS of the data. RMS is an indicator of energy content of a signal. As tool wear progresses, more energy is needed to drag the tool insert through the workpiece, it follows to reason that the RMS (or variance in this case) should increase. The variance is calculated using:

$$\sigma^2 = \frac{1}{T}\int_0^T x(t)^2 dt \tag{3.6}$$

  In equation 3.6 σ is the standard deviation. The variance is by definition the square of this. T is the time interval for which the integral is calculated. $x(t)$ is the signal for which the variance is calculated.

- *Skewness* is the third statistical moment and describes the distribution of the data in terms of symmetry or lack thereof, hence the term skewness. The skewness is calculated using:

$$S = \frac{1}{\sigma^3 T}\int_0^T x(t)^3 dt \tag{3.7}$$

- *Kurtosis* is the fourth statistical moment and is very popular in bearing condition monitoring. The kurtosis is a measure of the relative peakedness of the distribution, this is similar to the variance. The kurtosis is also a measure of how close the distribution is to the Gaussian distribution. It thus carries valuable information for condition monitoring. The kurtosis is calculated using:

$$K = \frac{1}{\sigma^4 T}\int_0^T x(t)^4 dt \tag{3.8}$$

- *Crest factor* is another feature which is widely used in bearing condition monitoring and is a measure of the impulsiveness of a vibration signal. A truly random signal has a crest factor generally less than 3. The crest factor is calculated using:

$$CF = \frac{X_{max}}{X_{rms}} \tag{3.9}$$

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

- *Entropy* is a measure of the uncertainty or disorder of of a given signal. One can intuitively see that a signal with a higher energy content, as in the case of a worn tool, will display more disorder. The entropy measure used was Shannon entropy which is often used in wavelet analysis. Shannon entropy is calculated using:

$$E = -\sum_{i=1}^{N-1} {x_i}^2 log({x_i}^2) \tag{3.10}$$

In 3.10, x_i is the value of x at time $t = i$. N is the number samples the feature is calculated for. This is the same as the time interval for the statistical features.

- *Dynamism* is a measure of the rate of change of a quantity. This feature also captures dynamic behaviour of a signal in a similar way to the crest factor. Dynamism was used for speech and music segmentation by Ajmera et al. (2003). Dynamism is calculated with:

$$D = \sum_{i=1}^{N} \Big[x_i - x_{i+1} \Big]^2 \tag{3.11}$$

Features in the frequency domain

Of the more salient features are usually those in the frequency domain. These features are directly connected to changes in the dynamic behaviour These are calculated from the one-sided power spectral density (PSD) using:

$$\Psi = \int_{fl}^{fh} S_x(f) df \tag{3.12}$$

In eq. 3.12 S_x is the one-sided PSD function and fl and fh are the frequency band for which this number is calculated. Ψ can increase, or decrease with increasing tool wear. The case where Ψ increases is where the cutting process changes from smooth cutting to a breakaway process. This causes an increase in vibration amplitudes. The case where Ψ decreases is where the dynamics of the process is altered so much by the change in the contact interaction caused by tool wear, that a shift in the peak occurs. When the peak starts to move out of the frequency band, the spectral energy decreases.

Håkansson et al. (2001) showed that the frequency bands that are most likely to show an increase, are those around the natural frequencies of the tool holder. Other characteristics of the cutting process, such as the chip forming frequency may also be monitored for signs of tool wear. On the whole there is not a method that can be used to predict which frequency bands are most likely to be useful in TCM. Allen and Shi (2001) suggested monitoring two frequency bands. A lower and a higher. The higher band then captures the natural frequencies of the system. Scheffer (2003), Lim (1993) and Jiang et al. (1987) have each derived their own frequency bands which were useful for their work. These band are also process specific and also dependant on cutting parameters.

In this study frequency bands are also derived. This was done by hand and was therefore more of an art than a science. The approach that was used to find these frequency bands was firstly a summing algorithm. This algorithm summed the PSD functions of the tool during its lifetime. Peaks on this summed PSD show the regions where the most energy in the system is at. The search for relevant peaks were then focused on these areas. Two types of peaks in the energy spectrum my be found in these high energy regions:

1. peaks that are insensitive to tool wear and subsequently do not significantly increase or decrease during the life of the tool.

2. peaks that grow with tool wear.

It is because of this that the selection of frequency bands has not yet been automated.

3.2.2 Feature selection

Having extracted a number of features from the data, one usually wishes to reduce the number of features. This is because not all the features are sensitive to tool wear. To do this Scheffer (2001) proposed that the correlation coefficient be used for this selection process.

It is assumed that the progression of tool wear over time can be approximated by a straight line with a arbitrary gradient. This was chosen to be 40°. The correlation coefficient for each feature and the theoretical tool wear is then calculated. The correlation coefficient is a measure that describes to what degree certain values of one signal occurs with certain other values of another signal. The correlation coefficient is calculated using:

$$\mathrm{corr}(X,Y) = \frac{\sum\limits_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\left[\sum\limits_{i=1}^{n}(x_i-\overline{x})^2\sum\limits_{i=1}^{n}(y_i-\overline{y})^2\right]^{1/2}} \tag{3.13}$$

X and Y are the two signals which are to be compared. $\overline{x}$ and $\overline{y}$ denote the mean values of the variables.

A value close to 1 is indicative that high values of one signal occurs with high values of the other signal. In the case where the correlation coefficient is close to -1, large values of one signal coincides with small values of the other signal.

Once the correlation coefficients have been calculated the highest ones can be chosen as the ones that carry the most information on tool wear. Correlation coefficients in the negative range are also very valuable because it guaranties the independence of features on each other. A combination of both was thus used for the recognition system.

3.2.3 Feature space reduction

From the theory of HMMs it has been implied that this technique uses 1-dimensional arrays for training and recognition. The theory of HMMs may be extended to use multidimensional arrays, but it was decided to use an existing HMM toolbox, feature space reduction is necessitated.

Dimensional reduction is a common technique in pattern recognition. These techniques reduce the dimensionality of the data for easier handling. According to Fugate et al. (2000) it is futile to expect good estimates from the tails of multidimensional data unless there is a very large amount of independent data available. This is what is referred to as "the curse of dimensionality.". The curse of dimensionality is simply that the amount of data required for training increases exponentially if the dimensionality is increased.

A simple and well known method namely, principal component decomposition was applied to the data. All the data is then projected onto the first principal component to reduce the dimensionality from N dimensions to 1 dimension. This was chosen conveniently in order to use the HMM toolbox directly on the application. If needed another set of HMM could be created. These HMM would then use the some of the other principal components. The output of the HMM committees could then be combined to form a more robust recognition. This study will use only one principal component to establish the technique.

The principal component analysis (PCA) is a standard function in the statistics toolbox for MATLAB that uses a singular value decomposition to calculate the principal components of a data matrix. The principal components can also be calculated as the eigenvectors of the covariance matrix of the feature space. The eigenvector with the highest corresponding eigenvalue will then be the unit vector of the first principal component. When the feature space is projected onto this vector it becomes the first principal component. The eigenvalues are then a measure of the total variance explained by each principal component.

3.2.4 Discretisation and construction

To accommodate the DHMM the dimensionally reduced feature is discretized into a number of levels. This is done with respect to the maximum and minimum values of the samples used for training. All the values that fall in-between these two values are rounded to the "nearest" level. These levels are similar to the bins in a histogram.

The observation sequences are constructed from the discretized feature vector. This is simply done by segmenting the feature vector into lengths of N consecutive samples. This number N is a parameter that determines how much temporal information is contained in the sequence. The strength of HMM recognition lie in these observation sequences.