Temporal Visualisation

MANMOHAN BALLAGAN

MSc by Research in Pattern Analysis and Neural Networks

ASTON UNIVERSITY

September 2002

Acknowledgements

I would like to thank my supervisor, Ian Nabney, for his help and guidance. Thanks to my associate supervisor Sudhir Jain and my external examiner Sameer Singh. Thanks to Iain Strachan for his GTM through time Matlab code. Thanks to Peter Tino for his useful discussions. Finally thanks to Harmesh Suniara for his advice on trading financial markets.

ASTON UNIVERSITY

Temporal Visualisation

MANMOHAN BALLAGAN

MSc by Research in Pattern Analysis and Neural Networks

Thesis Summary

The Generative Topographic Mapping is a probability density model which describes the distribution of data in a space of several dimensions in terms of a smaller number of latent (or hidden) variables. The standard GTM (generative topographic mapping) has been extended to model time series by incorporating it as the emission density in a hidden Markov model. This thesis studies the use of the Generative Topographic Mapping through time model for predicting regime shifts in financial market data. We looked at several aspects of the model, and trained it on different data sets and show the process of quantifying the information in the visualisation plot.

Keywords: GTM through time, GTM, hidden Markov models, Baum-Welch algorithm, Financial Markets

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Data Visualisation

Latent variable models represent the probability density of data in a space of several dimensions in terms of a smaller number of latent or hidden variables. The Generative Topographic Mapping (GTM) is a non-linear latent variable model. GTM provides a principled alternative to the widely used Self-Organising Map (SOM) of Kohonen (1982), and overcomes most of the significant limitations of the SOM.

An important application of latent variable models is to data visualisation. Many of the models used in visualisation are regarded as defining a projection from the D-dimensional data space onto a two-dimensional visualisation space. By contrast the GTM model is defined in terms of a mapping from the latent space into the data space. For the purposes of data visualisation, the mapping is then inverted using Bayes' theorem, giving rise to a posterior distribution in latent space.

The GTM model has been adapted to handle time series data. GTM in its standard form assumes that the data is generated by independent, identically distributed random variables. This is a poor approximation for time series. The GTM algorithm has been extended to handle time series by incorporating it as the emission density in a hidden Markov model. This extension is known as GTM through time.

1.2 Motivation for Project

The aim of the project is to investigate the use of the GTM through time model for predicting regime changes in financial market. This model has been chosen on the basis that it is suited to handling time series data and the visualisation aspect possibly allows the extraction of meaningful information from noisy financial data. In this thesis we will firstly look at the theory behind the model. We then conduct a series of experiments to correlate visualisation plots obtained by training the model on financial data with actual price movements in the data.

What do we mean by a regime change? We can define movement in a market in terms of an uptrend where prices are increasing, a downtrend where there is a decrease in prices and a sideways trend where prices are moving in a range (Murphy, 1999). Of course characterisation of a trend is dependent on the time window of data observed. A regime change is where a significant and sustained change in direction occurs. The examples shown below will elaborate this point.

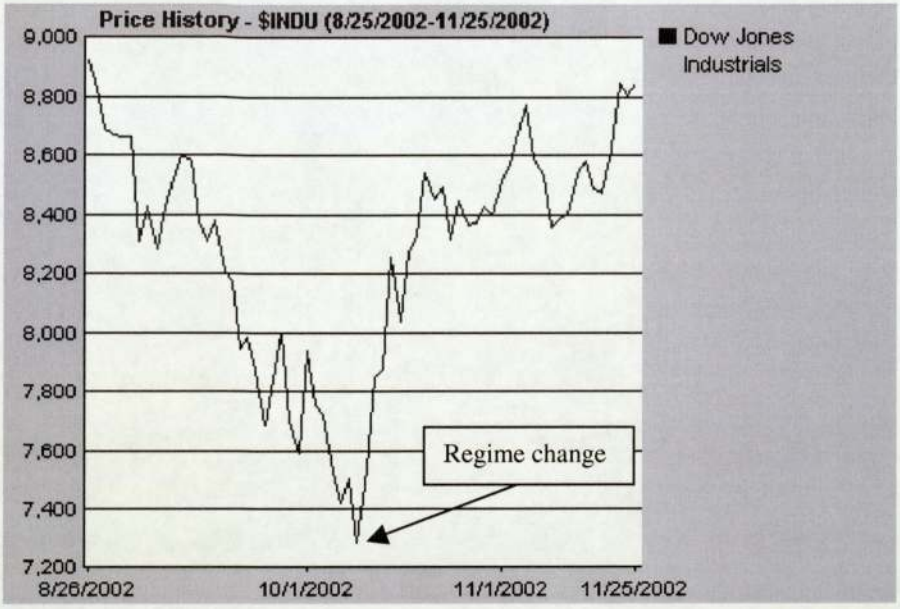


Figure 1.1 Dow Jones Index daily data from 25/8/2002-25/11/2002

In Figure 1.1 we can see three months of Dow Jones Index daily closing prices. There is a fairly clear downtrend in the first half of the chart with some small retracements upwards. Where the arrow points to we can see a significant reversal to an uptrend. This is a good example of a market in a downtrend then changing regime to an uptrend.

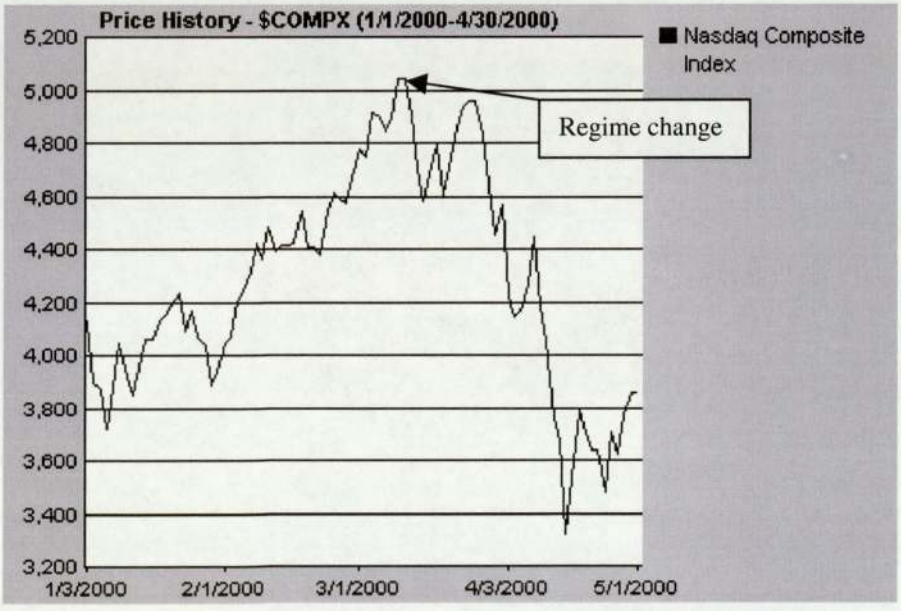


Figure 1.2 Nasdaq Index daily data from 1/1/2000-30/4/2000

In Figure 1.2 we can see three months of Nasdaq Index daily closing prices. There is an uptrend followed by a reversal to a downtrend although there is a reversal up before the downtrend continues. The arrow shows the point at which the regime change occurs.

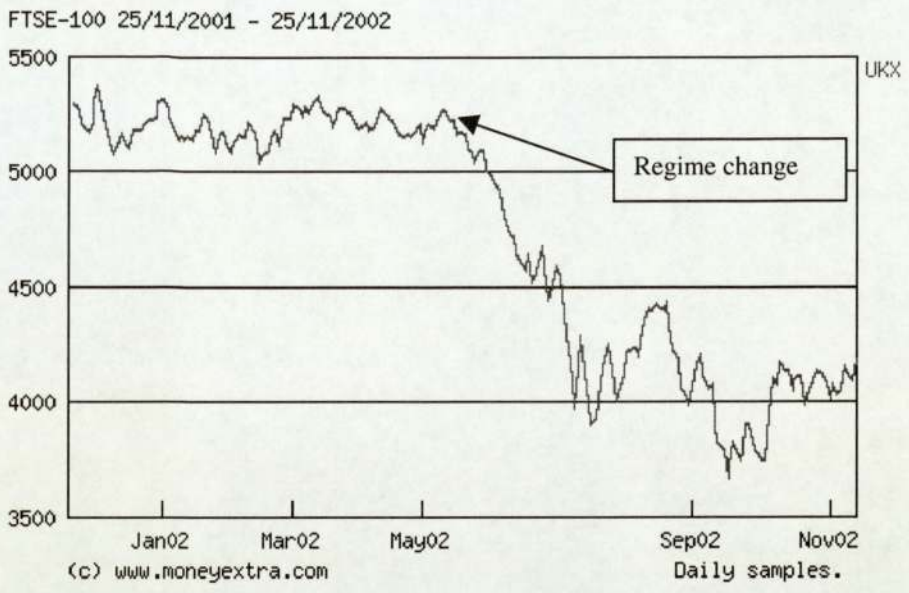


Figure 1.3 FTSE 100 Index daily data 25/11/2001 – 25/11/2002

In Figure 1.3 one year of FTSE 100 index daily data is displayed. In the first half of the chart the index moves in a narrow trading range and the market direction can be classed as sideways. At the point labelled by the arrow we can see the regime changes from sideways to a downtrend. Another type of change is a sideways market to an uptrend. Of course the reverse would also classify as a regime change where an uptrend or downtrend changes into a sideways market.

1.3 Aim of the project

In this project the GTM through time model is used to analyse financial market data. We wish to look for interesting behaviour exhibited by the model preceding significant regime changes. So we wish to have a prediction system giving us a signal before a regime change occurs. We will look at aspects of the model such as trajectories in the latent space plots, magnification factors and the probability of the data given the model and aim to correlate these with regime changes in the financial data.

Chapter 2

GTM Through Time

2.1 Introduction

The standard GTM algorithm assumes that the data on which it is trained consists of independent, identically distributed (i.i.d.) vectors. For time series the i.i.d. assumption is a poor approximation. In this chapter we show how the GTM algorithm can be extended to model time series by incorporating it as the emission density in a hidden Markov model. Since GTM has discrete hidden states we are able to find a tractable EM algorithm, based on the forward-backward algorithm, to train the model.

2.2 The Generative Topographic Mapping

We begin by reviewing the GTM algorithm for the standard case of i.i.d. data (Bishop, Svensen, Williams 1997). The goal of the GTM model is to find a representation for the distribution $p(\mathbf{t})$ of data in a D-dimensional space $\mathbf{t} = (t_1, ..., t_D)$ in terms L latent variables $\mathbf{x} = (x_1, ..., x_L)$. This is achieved by first considering a function $\mathbf{y(x;W)}$ which maps points $\mathbf{x}$ in the latent space into corresponding points $\mathbf{y(x;W)}$ in the data space. The mapping is governed by a matrix of parameters $\mathbf{W}$ and is represented by a radial basis network in which $\mathbf{W}$ represents the weights and biases. We are interested in the situation where the dimensionality L of the latent-variable space is less than the dimensionality D of the data space, since we wish to capture the fact that the data itself has an intrinsic dimensionality which is less than D. The transformation $\mathbf{y(x;W)}$ then maps the latent-variable space into an L-dimensional non-Euclidean manifold S embedded within the data space. This is illustrated schematically for the case of $L = 2$ and $D = 3$ in figure 2.1.

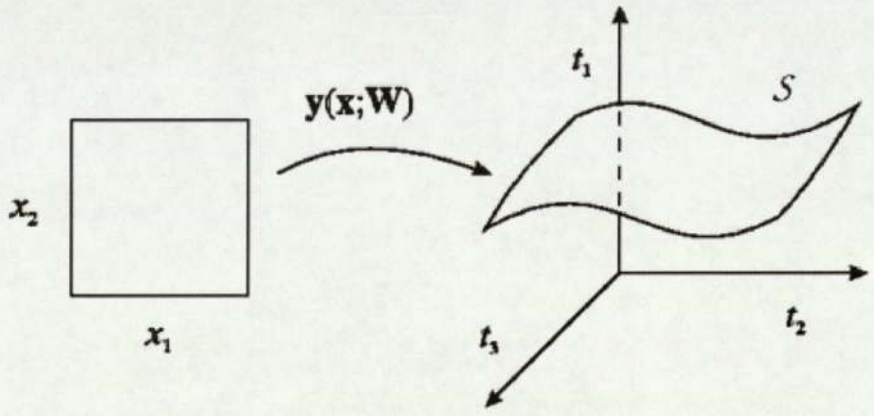


Figure 2.1: The non-linear function $\mathbf{y(x;W)}$ defines a manifold S embedded in data space given by the image of the latent-variable space under the mapping $\mathbf{x} \rightarrow \mathbf{y}$.

For the rest of this thesis L=2 as we are interested in analysing two-dimensional latent space plots.

If we define a probability distribution $p(\mathbf{x})$ on the latent-variable space, this will induce a corresponding distribution $p(\mathbf{y}\,|\,\mathbf{W})$ in the data space. We shall refer to $p(\mathbf{x})$ as the prior distribution of $\mathbf{x}$ for reasons which will become clear shortly. Since $L < D$, the distribution in $\mathbf{t}$-space would be confined to the L-dimensional manifold and hence would be singular. Since in reality the data will only approximately live on a lower dimensional manifold, it is appropriate to include a noise model for the $\mathbf{t}$ vector. We choose the distribution of $\mathbf{t}$, for given $\mathbf{x}$ and $\mathbf{W}$, to be a radially-symmetric Gaussian centred on $\mathbf{y(x;W)}$ having variance σ so that

$$p(\mathbf{t}\,|\,\mathbf{x},\mathbf{W},\sigma)=\left(\frac{1}{2\pi\sigma}\right)^{D/2}\exp\left\{-\frac{1}{2\sigma}\|\mathbf{y}(\mathbf{x};\mathbf{W})-\mathbf{t}\|^2\right\}. \tag{1}$$

The distribution in $\mathbf{t}$-space, for a given value of $\mathbf{W}$, is then obtained by integration over the prior distribution

$$p(\mathbf{t}\,|\,\mathbf{W},\sigma)=\int p(\mathbf{t}\,|\,\mathbf{x},\mathbf{W},\sigma)\,p(\mathbf{x})d\mathbf{x}. \tag{2}$$

For a given data set $\mathrm{D}=(\mathbf{t}_1,.....,\mathbf{t}_N)$ of N data points, we can determine the parameter matrix $\mathbf{W}$ and the variance σ using maximum likelihood. In practice it is convenient to maximise the log likelihood, given by

$$L\,(\mathbf{W},\sigma)=\ln\prod_{n=1}^{N}p(\mathbf{t}_n\,|\,\mathbf{W},\sigma). \tag{3}$$

Once we have specified the prior distribution $p(\mathbf{x})$ and the functional form of the mapping $\mathbf{y(x;W)}$, we can in principle determine $\mathbf{W}$ and σ by maximising $L\,(\mathbf{W},\sigma)$. However, the integral over x in (2) will, in general, be analytically intractable. If we choose $\mathbf{y(x;W)}$ to be a linear function of $\mathbf{W}$, and we choose $p(\mathbf{x})$ to be Gaussian, then the integral becomes a convolution of two Gaussians which is itself a Gaussian. For a noise distribution $p(\mathbf{t}\,|\,\mathbf{x})$ which is Gaussian with a diagonal covariance matrix, we obtain the standard factor analysis model. In the case of the radially symmetric Gaussian given by (1) the model is closely related to principal component analysis since the maximum likelihood solution for $\mathbf{W}$ has columns given by the scaled principal eigenvectors. Here we wish to extend this formalism to non-linear functions $\mathbf{y(x;W)}$, and in particular to develop a model which is similar in spirit to the SOM algorithm. We therefore consider a specific form for $p(\mathbf{x})$ given by a sum of delta functions centred on the nodes of a regular grid in latent space

$$p(\mathbf{x})=\frac{1}{K}\sum_{i=1}^{K}\delta(\mathbf{x}-\mathbf{x}_i) \tag{4}$$

in which case the integral in (2) can be performed analytically. Each point $\mathbf{x}_i$ is then mapped to a corresponding point $\mathbf{y(x}_i\mathbf{;W)}$ in data space, which forms the centre of a Gaussian density function, as shown in figure 2.2 below.

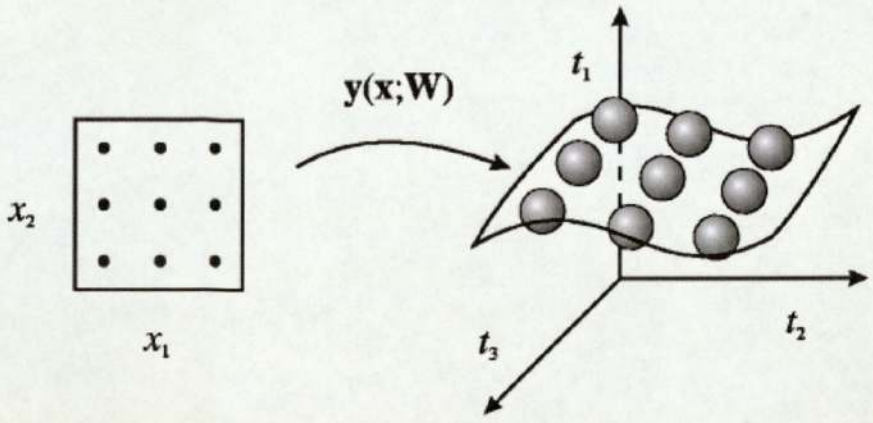


Figure 2.2: Each point $\mathbf{x}_i$ in the latent space is mapped to a corresponding point $\mathbf{y}(\mathbf{x}_i;\mathbf{W})$ in data space, and forms the centre of a corresponding Gaussian distribution.

From (2) and (4) we see that the distribution function in data space then takes the form

$$p(\mathbf{t}\,|\,\mathbf{W},\sigma) = \frac{1}{K}\sum_{i=1}^{K} p(\mathbf{t}\,|\,\boldsymbol{x}_i,\mathbf{W},\sigma) \tag{5}$$

and the log likelihood function becomes

$$L\,(\mathbf{W},\sigma) = \sum_{n=1}^{N} \ln\left\{\frac{1}{K}\sum_{i=1}^{K} p(\mathbf{t}_n\,|\,\mathbf{x}_i,\mathbf{W},\sigma)\right\}. \tag{6}$$

For the particular noise model $p(\mathbf{t}\,|\,\mathbf{x},\mathbf{W},\sigma)$ given by (1), the distribution $p(\mathbf{t}\,|\,\mathbf{W},\sigma)$ corresponds to a constrained Gaussian mixture model since the centres of the Gaussians, given by $\mathbf{y}(\mathbf{x}_i;\mathbf{W})$, cannot move independently but are related through the function $\mathbf{y}(\mathbf{x};\mathbf{W})$. Note that, provided the mapping function $\mathbf{y}(\mathbf{x};\mathbf{W})$ is smooth and continuous, the projected points $\mathbf{y}(\mathbf{x}_i;\mathbf{W})$ will necessarily form a topographic mapping in the sense that any two points $\mathbf{x}_A$ and $\mathbf{x}_B$ which are close in latent space will map to points $\mathbf{y}(\mathbf{x}_A;\mathbf{W})$ and $\mathbf{y}(\mathbf{x}_B;\mathbf{W})$ which are close in data space.

For data vectors $\mathbf{t}_n$ which take the form of a time series it is no longer appropriate to assume that the vectors are independent. Typically, vectors corresponding to nearby times will be highly correlated. Such effects can be captured using the Hidden Markov model (HMM) formalism (Rabiner 1989). GTM can be extended within the HMM framework to represent temporal data and the model is known as GTM Through Time (Bishop, Hinton, Strachan 1997).

2.3 GTM Through Time

See the Appendix for a detailed description of Hidden Markov Models. The structure of the GTM through time model is illustrated in figure 2.3 below, in which the hidden states of the model at each time step are labelled by the index l corresponding to the latent points $\{\mathbf{x}_l\}$. We introduce a set of transition probabilities p_{ij} corresponding to the probability of making a transition to state j given that the current state is i. The emission density for the hidden Markov model is then given by the GTM density model. It should be noted that both the transition probabilities p_{ij} and the parameters $\mathbf{W}$ and σ governing the GTM model are common to all time steps, so that the number of adaptive parameters in the model is independent of the length of the time series. We also allow separate prior probabilities p_l on each of the latent points at the first time step of the algorithm.

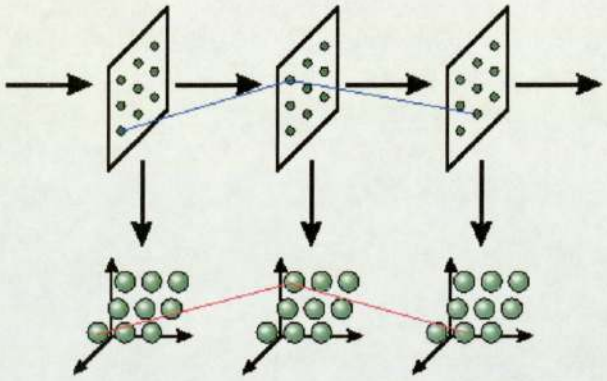


Figure 2.3: The upper half of the diagram shows a two dimensional latent space and lower half shows a three dimensional data space. The GTM through time model is shown in generative mode (so we know which latent point is responsible for each data point) and there are stochastic transitions between latent states. Note that the parameters of the GTM model, as well as the transition probabilities between states, are tied to common values across all time steps.

In figure 2.3 the top half of the diagram shows the latent point grid and the bottom half shows the corresponding Gaussians as we move through time left to right. So as we move through time the hidden state of the model changes and this is indicated by a blue line. The corresponding Gaussian in data space also changes and this is indicated by a red line. At each point in time the model can generate a data point from the selected Gaussian density function. The model is shown in *generative mode* in the diagram where the Gaussian responsible for each data point is known.

Training mode - The idea of training the GTM through time model is to regard the identity of the Gaussian responsible for each data point as a missing variable and use the EM algorithm to maximise the likelihood. The parameters of the model are changed during training. The E-step is used to compute the probabilities of the state variables z_{ni} and $\xi_n(i,j)$ which represents the joint posterior probability of being in state i at time n and state j at time $n+1$ using the forward-backward algorithm (Section 2.4). This is coupled with the transition and emission probabilities in equation (9) to give the complete data likelihood. The values of $\xi_n(i,j)$ are used in the M-step to find the mixing coefficients and weights.

Inference mode (generalisation) - Here we also calculate the posterior probabilities of the states (using forward-backward), and average these to give a single point for the visualisation plot. This is done on test data. So as we move through time we are in effect jumping from one Gaussian to another and then inverting the transformation from latent space to data space to produce a trajectory in the latent space. For visualisation we use the E-step to calculate the responsibilities of each Gaussian for each data point. This produces a distribution for each data point and we plot the mean. Visualisation is discussed in further detail in section 2.5.

If we use a fully connected matrix of independent transition probabilities connecting every state at time n to every state at time $n+1$, then the number of independent parameters would be prohibitively large. If we have, for example, 100 hidden states in the GTM model (a relatively small number) then we would have 10^4 independent transition probability parameters to be determined (slightly less in fact due to the constraint that probabilities must sum to one). This would require an excessive amount of training data.

Also it fails to capture any prior knowledge which we might possess about the nature of the transitions between different time steps. In many applications we expect different regions of the latent space to correspond to different regimes. We also expect smooth changes in latent space within a regime and relatively rare jumps to other regimes. An approximate way to capture this knowledge is to allow groups of transitions to be governed by a common parameter. So for example suppose we have 64 points in the latent space.

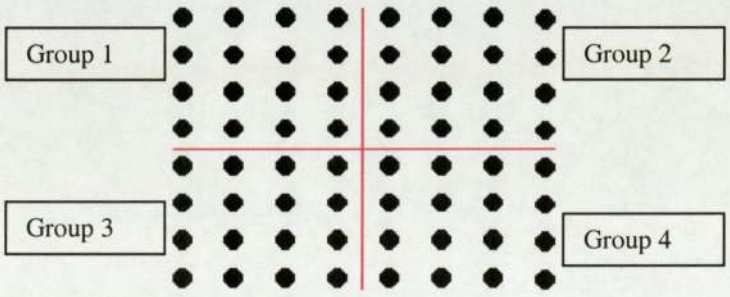


Figure 2.3a: Latent space grid split into four groups.

In figure 2.3a we can see a grid of 64 latent points split into four groups as labelled. Each quadrant corresponds to a different group. We first calculate the probability transition matrix based on the responsibility of each Gaussian for each data point. We then calculate the transition probabilities of going from each latent point i to latent points in each group k and denote these by η_{ik}, and these satisfy $\sum_k \eta_{ik} = 1$. We denote the kth group by G_k and we introduce indicator variables C_{kj} which equal 1 if state j is in group G_k and 0 otherwise. The transition probability from state i to state j is then given by

$$p_{ij} = \sum_k \eta_{ik} C_{kj} N_k^{-1}$$

where N_k denotes the number of states in group G_k.

2.4 EM Algorithm

The model is trained using a set of N data vectors $\mathbf{t}_1,.....,\mathbf{t}_N$ in which the parameters $\mathbf{W}$ and σ, as well as the transition probabilities, are determined by maximum likelihood. To derive the correct likelihood function we note that the model represents a generative distribution for time series data as follows. At the first time step we select a latent point i with probability π_i and then generate the first data vector $\mathbf{t}_1$ by sampling from the corresponding Gaussian component $p(\mathbf{t}|\mathbf{x}_i)$ of the GTM model. Next we make a transition to a new state j with probability p_{ij} and again generate a data point from the corresponding component $p(\mathbf{t}|\mathbf{x}_j)$. From this we see that the likelihood function for a given observed sequence of vectors $\mathbf{t}_1,.....,\mathbf{t}_N$ can be written

$$\sum_{i_1}...\sum_{i_N}\pi_{i_1}p(\mathbf{t}_1|\mathbf{x}_{i_1})p_{i_1i_2}p(\mathbf{t}_2|\mathbf{x}_{i_2})....p_{i_{n-1}i_n}p(\mathbf{t}_N|\mathbf{x}_{i_N}) \tag{7}$$

where i_n denotes the state at step n. The summations correspond to a sum over all possible trajectories through the hidden states of the model. At first sight it would therefore appear that the evaluation of (7) would be an extremely complex undertaking since the number of paths through the hidden states grows exponentially with N. However, because of the discrete nature of the hidden states, we can obtain an efficient algorithm for training this model.

We can regard the identity of the component responsible for generating each data point as a missing variable, and use the EM (expectation-maximisation) algorithm to maximise the likelihood. In the context of hidden Markov models this is generally known as the Baum-Welch algorithm. To obtain the EM algorithm for this model we first introduce a set of binary indicator variables z_{ni} which denote the state i of the system at step n. We shall regard the z_{ni} as missing variables. If the z_{ni} were given, then the complete-data likelihood would take the form

$$L_c=\prod_{n=1}^{N-1}\prod_{i_n}\left\{\pi_{i_1}p(\mathbf{t}_1|\mathbf{x}_{i_1})\right\}^{z_{1i_1}}\left\{p_{i_n,i_{n+1}}p(\mathbf{t}_{n+1}|\mathbf{x}_{i_{n+1}})\right\}^{z_{ni_n}z_{ni_{n+1}}}. \tag{8}$$

The algorithm involves first making an initial guess for the parameters $\mathbf{W},\sigma$ and η_{ik}. We next take the expectation of the logarithm of the complete-data log likelihood function (8) with respect to the posterior distribution of the z_{ni} (evaluated using the current values of the parameters), and use $\langle z_{ni}z_{nj}\rangle=\xi_n(i,j)$, where $\xi_n(i,j)$ denotes the joint posterior probability of being in state i at time n and state j at time $n+1$, to give

$$\langle \ln L_c\rangle=\sum_{n=1}^{N-1}\sum_{i_n}\xi_n(i_n,i_{n+1})\ln\left\{p_{i_n,i_{n+1}}p(\mathbf{t}_n|\mathbf{x}_{i_n})\right\}. \tag{9}$$

The posterior probabilities $\xi_n(i,j)$ are obtained in the E-step using the standard forward backward algorithm (See Appendix). The transition probability of state in group i to a state in group k is denoted

by η_{ik} . Maximising (9) with respect to the η_{ik} and using a Lagrange multiplier to enforce the constraint $\sum_k \eta_{ik} = 1$ we obtain

$$\eta_{ik} = \frac{\sum_n \sum_{j \in G_k} \xi_n(i,j)}{\sum_k \sum_n \sum_{j \in G_k} \xi_n(i,j)}. \tag{10}$$

Similarly we can maximise (9) with respect to $\mathbf{W}$ to obtain the M-step equation

$$\mathbf{\Phi}^{\mathrm{T}} \mathbf{G}_{old} \mathbf{\Phi} \mathbf{W}_{new}^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}} \mathbf{R}_{old} \mathbf{T} \tag{11}$$

where $R_{in} = \sum_j \xi_n(i,j)$ denotes the posterior probability of state i at step n, $\mathbf{\Phi}$ is a $K \times M$ matrix representing the RBF mapping with elements $\Phi_{ij} = \phi_j(\mathbf{x}_i)$, $\mathbf{T}$ is a $N \times D$ data matrix with elements t_{nk}, $\mathbf{R}$ is a $K \times M$ matrix with elements R_{in}, and $\mathbf{G}$ is a $K \times K$ diagonal matrix with elements

$$G_{ii} = \sum_{n=1}^{N} R_{in}(\mathbf{W}, \sigma). \tag{12}$$

We can solve (11) for $\mathbf{W}_{new}$ using standard matrix inversion techniques based on singular value decomposition to allow for possible ill-conditioning. Note that the matrix $\mathbf{\Phi}$ is constant throughout the algorithm and so need only be evaluated once at the start.

Finally, maximising (9) with respect to σ gives

$$\sigma_{new} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in}(\mathbf{W}_{old}, \sigma_{old}) \left\| \mathbf{y}(\mathbf{x}_i; \mathbf{W}_{new}) - t_n \right\|^2 .$$

After a complete M-step the new parameter values are used in the next E-step to re-evaluate the posterior probabilities, and so on to convergence.

2.5 Application to visualisation

To apply GTM through time to data visualisation, Bayes' theorem is used to invert the transformation from latent space to data space. For the particular choice of prior distribution given by (4), the posterior distribution is again a sum of delta functions centred at the lattice points $\boldsymbol{x}_i$, with coefficients given by responsibilities R_{in}; the probability of the i'th gaussian generating the n'th data point. These coefficients can be used to provide a visualisation of the posterior responsibility map for individual data points in the two-dimensional data space. In this project we wish to visualise a set of data points representing a time series so a complete posterior distribution $p(\boldsymbol{x} \mid \boldsymbol{t}_n)$ for each data point may provide too much information so we summarise the posterior by its mean, given for each data point $\mathbf{t}_n$ by

$$\begin{aligned} \langle \mathbf{x} \mid \mathbf{t}_n, \mathbf{W}, \sigma \rangle &= \int p(\mathbf{x} \mid \mathbf{t}_n, \mathbf{W}, \sigma) \mathbf{x} \, d\mathbf{x} \\ &= \sum_{i=1}^{K} R_{in} \mathbf{x}_i . \end{aligned}$$

So for each data point we plot the posterior mean in latent space and this forms a trajectory of points when we do this for all data points.

2.6 Magnification factors

GTM is a powerful visualisation tool but there are some aspects of the data structure that it does not show clearly in its standard form. In particular, even if the data consists of well-separated clusters of points, the latent space representation will be much closer to a uniform distribution because of the choice of prior distribution.

It is easy to see why this should be so if we consider the Gaussian mixture model on the GTM manifold. The EM algorithm will attempt to place the mixture components in regions of high data density and will move the components away from regions of low data density. It can do this because the non-linear map from latent space to data space enables the manifold to stretch across regions of low data density. This stretching (or magnification) can be measured using techniques of differential geometry, and plotting the magnification factors in latent space allows the user to see separation between clusters (Nabney, 2001).

Consider a rectangular Cartesian set of co-ordinates x_i for $i = 1,.....,L$ in the latent space. Under the smoothly differentiable RBF mapping, these are transformed to a set of L coordinates ς^i on the L-dimensional manifold M . To determine the magnification factor, we need to work out the change in a small volume dV in latent space[1] mapped to a small volume dV' on M . The volume dV is infinitesimal, so we shall consider a hypercuboid at a point $\boldsymbol{p}$ in latent space (a square for $L = 2$) whose sides are aligned with the latent variable axes. This is mapped, up to first order, to a L-dimensional parallelepiped (a parallelogram for $L = 2$) at a point $\boldsymbol{p}' = \boldsymbol{y}(\boldsymbol{p};\boldsymbol{W})$ in the data space whose sides are given by the tangent vectors to the curvlinear coordinates ε^i at $\boldsymbol{p}'$, i.e $(\partial \boldsymbol{y}/\partial x_i)dx_i$.

We denote by $\boldsymbol{J}$ the $L \times D$ Jacobian matrix of the map $\boldsymbol{y}(\boldsymbol{x};\boldsymbol{W})$:

$$\boldsymbol{J} = (J_{kl}) = \frac{\partial y_k}{\partial x_l}.$$

The volume of a D-dimensional parallelepiped is equal to the determinant of the vectors along its sides expressed with respect to a D-dimensional basis. However the sides of dV' are given by the L rows of $\boldsymbol{J}$ in a D-dimensional space. Let V_P denote the vector space spanned by the rows of $\boldsymbol{J}$; we can find an orthogonal basis B for this space by the Gram-Schmidt process. Let the $L \times D$ matrix $\boldsymbol{M}$ contain this basis as its columns, and compute

$$\hat{\boldsymbol{J}} = \boldsymbol{J}\boldsymbol{M} .$$

[1] We are mainly interested in the case L=2, when we can replace 'volume' by 'area'.

Since $\boldsymbol{M}$ is a projection matrix, it follows that the rows of $\hat{\boldsymbol{J}}$ are the same vectors as the rows of $\boldsymbol{J}$ but expressed with the respect to the basis B . Hence the volume dV' is equal to det $\hat{\boldsymbol{J}}$, which can be computed since $\hat{\boldsymbol{J}}$ is a square $L \times L$ matrix.

However, we can avoid having to find the matrix M by the following observation:

$$\hat{\boldsymbol{J}}\hat{\boldsymbol{J}}^T = \boldsymbol{J}\boldsymbol{M}\boldsymbol{M}^T\boldsymbol{J}^T = \boldsymbol{J}\boldsymbol{J}^T .$$

Then using this result and the properties of determinants:

$$(\det(\hat{\boldsymbol{J}}))^2 = \det(\hat{\boldsymbol{J}})\det(\hat{\boldsymbol{J}}) = \det(\hat{\boldsymbol{J}})\det(\hat{\boldsymbol{J}}^T) = \det(\hat{\boldsymbol{J}}\hat{\boldsymbol{J}}^T) = \det(\boldsymbol{J}\boldsymbol{J}^T) .$$

But $\boldsymbol{J} = \boldsymbol{\psi}\boldsymbol{W}$ where $\boldsymbol{\psi}$ has elements $\psi_{ji} = \partial\phi_j / \partial x^i$. Hence the magnification factors are given by

$$\frac{dV'}{dV} = \det{}^{1/2}(\boldsymbol{\psi}\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{\psi}^T) .$$

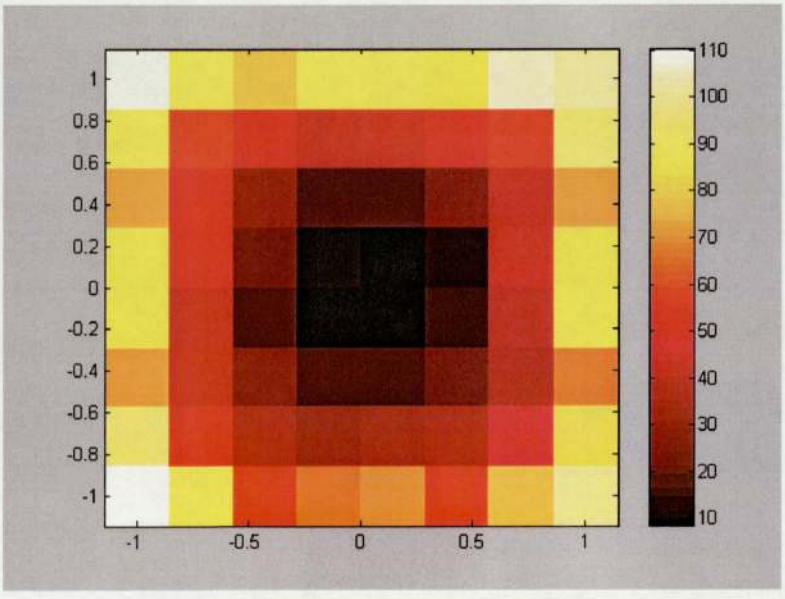


Figure 2.4: Magnification factors shown by shading of latent grid squares.

In figure 2.4 we can see a latent space plot with magnification factors shown. The posterior means corresponding to the data points are not shown. There are lower magnification factors in the centre of the plot and higher values around the perimeter. The magnification factors give us an indication of distance between data points in data space where larger magnification factors correspond to a larger distance. In the context of this project we are looking for interesting behaviour leading up to regime changes. We will be looking at changes in magnification factors to find a correlation between changes in regime and changes in magnification factors along the latent space trajectory.

2.7 Summary

In this chapter we have reviewed the standard GTM algorithm. With the groundwork complete we discussed GTM through time as an extension of GTM. Finally, the concept of magnification factors was explained. In the following chapter we look at the practical implementation of the model and visualisation in the latent space.

Chapter 3

Model Implementation

3.1 Introduction

The aim of the project is prediction of regime shifts in financial market data. To achieve this, we look for interesting behaviour in the latent space plots leading up to and during regime shifts, such as movement from one region of the latent space to another. Changes in magnification factors will also be considered.

In this chapter we first look at the data and how it is manipulated before being input to the GTM through time model. We then look at the model training algorithm. Finally, we look at a latent space plot representing a time series.

3.2 Data and Model Implementation

3.2.1 Data used

When considering a financial instrument such as a stock or an index we look at key values that are recorded for that instrument, such as the daily opening price, high and low for the day and the closing price. The closing price is regarded as the key measure and this is used in this project when considering both daily and intraday data. For intraday data, for example on a five minute chart, the closing price is the price at the end of a five minute period of trading so that in one hour we would have twelve closing price values.

The daily data used in all the experiments was obtained from a CD that is supplied with the software package Tradestation developed by Omega Research. The intraday data was obtained from a Bloomberg data feed.

Index, currency and stock data was used in the experiments performed and this is shown in Table 3.1, with a description of each item.

Dow Jones Index daily data.	Key US index comprising 30 companies.
Intel daily data.	Key US technology stock.
Dollar/Pound daily data.	US dollar rate versus British pound.
Dollar/Yen daily data.	Most traded currency in the world in dollar terms.
Dow 5 minute data.	Intraday Dow Jones Index data.
S&P 500 daily data.	Key US index comprising 500 companies.
Dollar Index daily data	The dollar weighted against a basket of currencies

Table 3.1: Data used in experiments

The data used was selected both due to availability and the fact the markets chosen are key economic indicators in the US and indeed worldwide.

3.2.2 Delay Vectors

Much of non-linear time series analysis is based on embedding vectors, i.e., the construction and manipulation of delay vectors. Given a long series of data, one creates a group of short data samples from it, subseries of length m called "delay vectors". One then considers this series of delay vectors as a time series in itself, and analyses the dynamics of this derived time series, under the assumption that it will mimic the dynamics of the complex system that gave rise to the original time series under analysis (Ben Goertzel, 1998). We use singular value decomposition to decide the dimension of the delay vectors.

An $m \times n$ matrix $\mathbf{A}$ can be written in the form

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and Σ is a diagonal matrix with the same dimension as $\mathbf{A}$. The elements of Σ are the singular values. The reason for looking at plots of the singular values is Taken's Embedding Theorem (Noakes, 1991). The idea is that if the delay vector is sufficiently long, then we get a diffeomorphism of the manifold represented in the underlying system variables (which are unknown) to the delay vector space. What we are trying to work out is how large the delay vectors should be for this diffeomorphism to hold. The singular values give the length of the axes of the manifold, so the idea is that while making the delay vector longer, once these lengths have stabilised the underlying manifold is staying the same, and hence there is no need to expand the delay vectors. The problem is that this assumes that the time series is stationary (which it rarely is for financial time series). In a non-stationary environment we have to be a little more relaxed about the spectrum stabilisation. This is where it does become rather more subjective. It also explains why we are interested in the larger singular values, since the smaller ones are (we assume) generated by noise processes that we would rather not model.

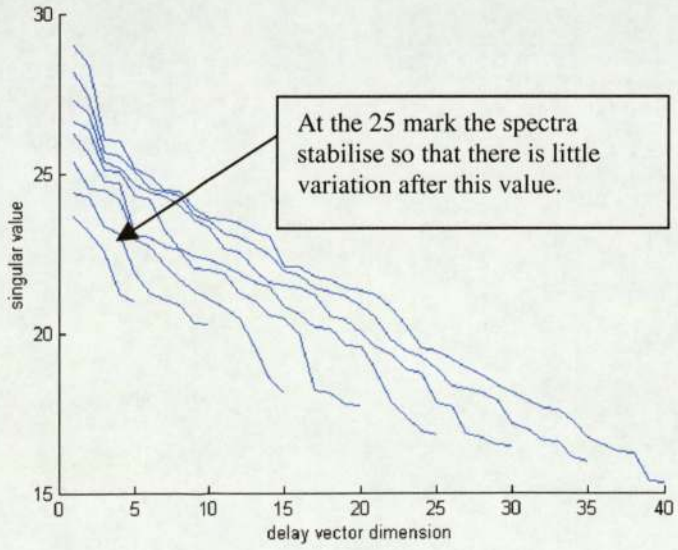


Figure 3.1 Delay vector spectra for Dow Jones daily data. A stabilising of the spectra can be observed for dimension >= 25.

Spectra were plotted for different delay vector dimensions. Each line gives the singular value spectrum for a certain length of delay vector; the number of singular values is equal to the length of the vector, so a delay vector of length 10 has 10 singular values, while one of length 25 has 25. A stabilising of the spectra indicates an appropriate dimension for the delay vectors. In figure 3.1 spectra for Dow Jones daily data are displayed. We can see that a value of 25 is appropriate for delay vector dimension.

3.2.3 Other parameters in model

There are parameters in the model that have to be adjusted by hand (Nabney, 2001). These are the number and type of the basis functions in the radial basis function (RBF) (Bishop, 1995) and the number and distribution of the latent space sample points. The RBF basis function parameters control the complexity, or smoothness, of the map from latent space to data space. For example if Gaussian basis functions are used, then the ratio of their standard deviation to the spacing of the basis functions affects the curvature of the manifold in data space. The optimal number of latent space sample points and basis functions was determined by the experiment described in Section 3.2.4. If there are too few sample points compared to the number of basis functions, then the Gaussian components in data space become relatively independent and there is effectively no manifold.

3.2.4 Selection of model parameters

In order to choose an optimal set of parameters the model was trained on 500 points of Dow Jones daily data and then a further 300 points were used for validation. The model parameters varied as shown in Table 3.2 below and the error values obtained were recorded.

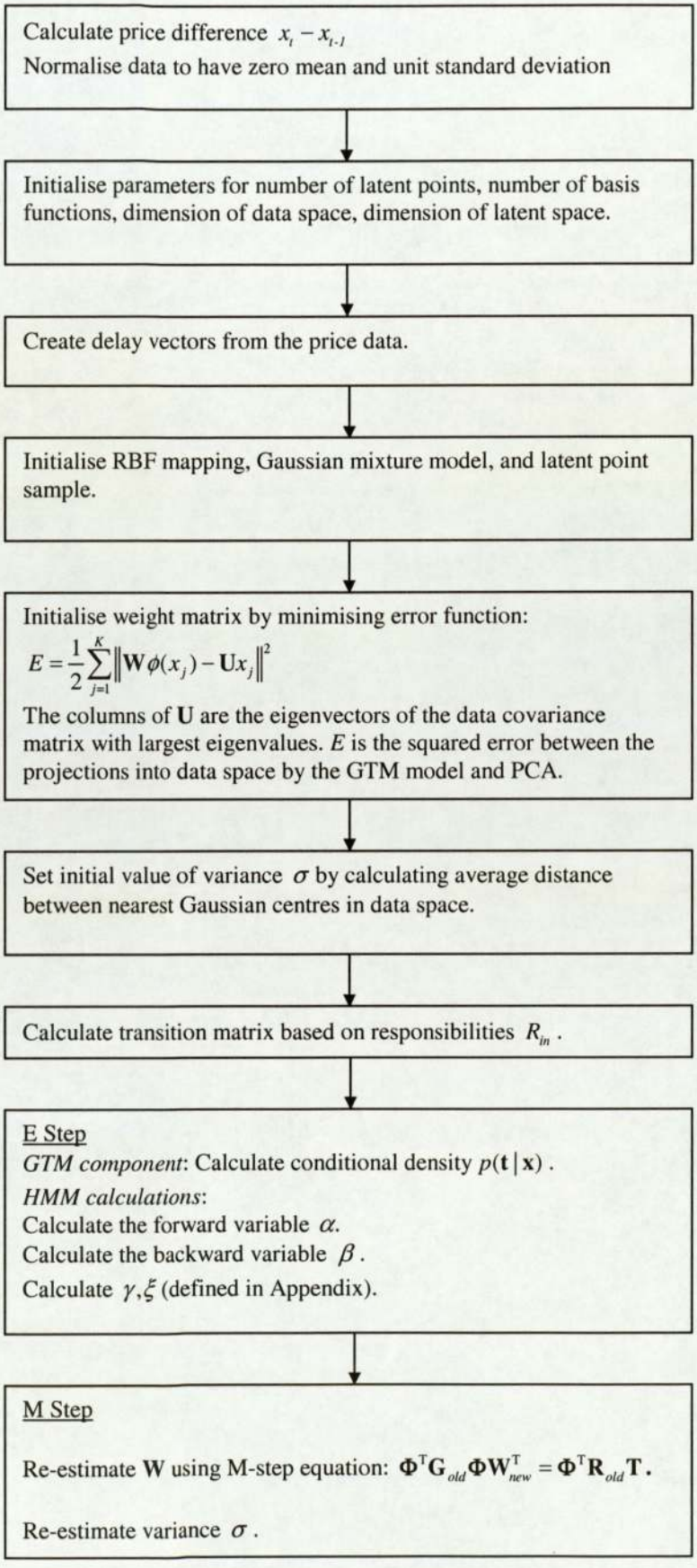
Number of Latent Points	Number of Basis Functions	Error value after 30 cycles	Negative log likelihood per point (training set)	Negative log likelihood per point (validation set)	Time taken for training/s
16	4	6528.91	13.32	20.36	2.03
36	4	6475.42	13.21	20.46	3.46
64	4	6448.49	13.15	20.57	6.20
16	16	6145.59	12.54	20.54	2.64
36	16	6079.29	12.40	20.79	3.41
64	16	6008.57	12.26	20.78	6.32
16	36	6050.87	12.34	20.35	1.98
36	36	5688.10	11.60	21.40	3.46
64	36	5578.37	11.38	21.79	6.43

Table 3.2: Results of parameter optimisation experiment.

Increasing the number of latent points to 100 caused convergence problems so this is not shown in the table. We deduce an optimal set of parameters by looking at the lowest negative log likelihood values for the validation set. From the results above choosing number of latent points equal to 16 and number of basis functions equal to 4 or 36 are optimal choices. As there are relatively few training points we choose 4 as the number of basis functions. However, for visualisation purposes it is beneficial to increase the number of latent points to 64 even though the validation likelihood is slightly suboptimal.

3.2.5 Training algorithm

The training algorithm is outlined as follows:



The E and M steps are repeated for a fixed number (30) cycles.

3.2.6 Coding of model

Code written by Iain Strachan implementing the GTM through time model was available on the NCRG server. This code was restructured and added to as necessary.

3.2.7 Analysing the plot in latent space

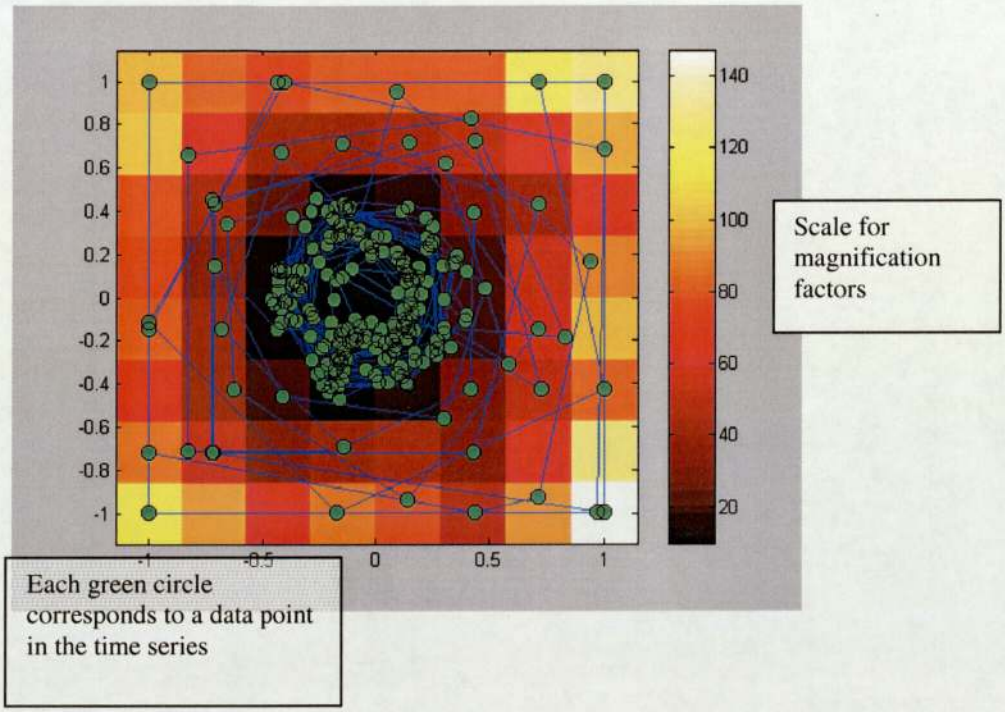


Figure 3.2: Latent space plot of Dow Jones Index daily data.

In Figure 3.2 a typical latent space plot is displayed. The coloured squares represent the magnification factor at each latent point. There are 64 latent points in this grid. Each green circle is the posterior mean for a point in the original time series. The green circles are connected by lines to produce a trajectory of points and it is the characteristics of this trajectory we are interested in. So, for example, large jumps from one point to the next were studied to see if these correspond to significant price changes. We can see that magnification factors are lower in the centre of the plot and higher around the perimeter. So movements from a low magnification area to a high magnification area were studied to see if these correspond to regime shifts in the source data. The magnification factors shown in the latent space plot are for visualisation purposes but for a precise measurement we calculate the distance between data points in data space.

A set of interactive tools was developed to allow us to step through the latent space trajectory and analyse the corresponding price movement.

An initial set of exploratory experiments was conducted to explore aspects of the model in terms of visualising financial data. These experiments formed a basis for more rigorous experiments conducted later (See Chapter 5).

3.2.8 Initial Transition Probabilities

In the original code, the initial transition probability matrix contained random values. The transition matrix initialisation was altered as follows. The responsibilities of each Gaussian for each data point are calculated. Each data point is assigned to the Gaussian (and hence latent space point) with maximum responsibility for it. The transitions between latent space points are then used to calculate the transition matrix. This transition matrix is then split into four equal grids; each grid represents a grouping of latent points. The transition matrix is altered based on this grouping structure and the calculation is discussed in detail in section 2.3.

The GTM through time model was trained on Dow Jones Index daily data using both the random and non-random initial transition matrices and the calculations were timed. The results of the calculations are shown in Table 3.3.

Non random transition matrix (10 experiments performed)
Error = -1231.96 after 15 cycles
Average time taken for transition matrix calculation = 0.473
Average time for EM algorithm = 1.132

Random transition matrix (10 experiments performed)
Random: Error = - 1233.03 after 7 cycles
Average time taken for transition matrix calculation = 0.043
Average time for EM algorithm = 2.807

Table 3.3: Comparison of random and non-random transition matrix calculations.

Discussion of results

For the random transition matrix calculation, the EM algorithm converged to a stable value after 7 cycles and the average time taken for training was 1.132 seconds. The average time taken for calculation of the transition matrix was 0.043 seconds.

For the non-random transition matrix calculation, the EM algorithm converged to a stable value after 15 cycles and the average time taken for training was 2.807 seconds. The average time taken for calculation of the transition matrix was 0.473 seconds.

From these results the random transition matrix appears to be more efficient as the calculation time is an order of magnitude faster than the non-random method. However the non-random transition matrix more accurately represents the structure of the training data. The non-random method is used in all other experiments.

3.3 Summary

In this chapter we have covered the practical implementation of the GTM through time model. We have also discussed the data we will use in the experiments and how we use the method of delay vectors. We also looked at a sample latent space plot and how we will be using this to study regime changes in the underlying data. In the next chapter we look at a range of exploratory experiments.

Chapter 4

Exploratory Experiments

4.1 Introduction

In this chapter we discuss a range of experiments that were performed to use the GTM through time model to detect regime changes in financial data. Many of the experiments failed to provide any useful conclusions regarding visualisation in the latent space and regime changes but are discussed for completeness.

4.2 Generalisation Performance

When proceeding from training the model on a set of data to running the model forward on test data the E-step of the forward-backward calculation produced ill conditioned matrices unless the data was pre-processed using either price difference or log ratios. So generalisation to new data requires that the data be pre-processed in this way. In all the following experiments the method of price difference was used – the choice was arbitrary given that the latent space plots obtained are very similar.

The characteristics of the trajectory of pre-processed data in the latent space are completely different to those of the trajectory obtained using raw data.

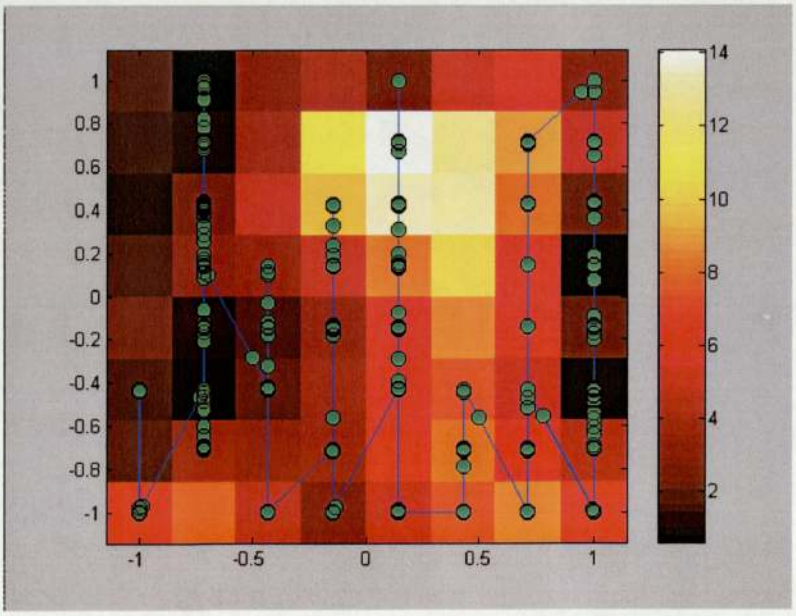


Figure 4.1: Latent space plot of posterior means using raw training data

In Figure 4.1 we can see the latent space plot produced by training the model on two years of Dow Jones Index daily data which has not been pre-processed – compare this with Figure 4.1a. As the model does not generalise using raw data, this type of latent space plot will not be considered further.

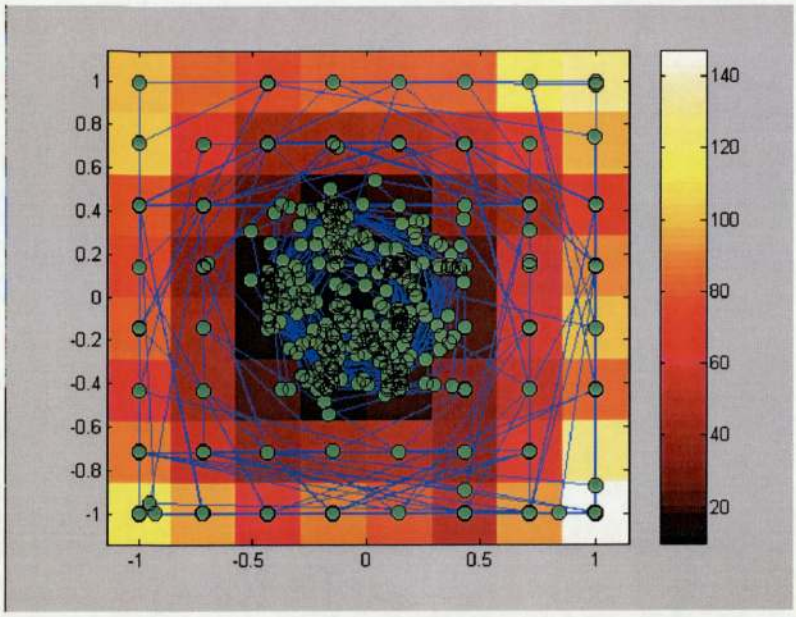


Figure 4.1a: Latent space plot of posterior means using pre-processed training data

In Figure 4.1a we can see the latent space plot produced by training the model on two years of Dow Jones Index daily data which has been pre-processed using price difference.

The model was trained on daily Dow Jones Index data from 1996-1997 and then tested on all of the points for 1998 so we are looking at adjacent time periods. The number of latent points was equal to 64 and the number of basis functions was 4. This choice was based on the validation experiments discussed previously in section 3.2.4. The number of training points was approximately 500 with 250 test data points. The training data was normalised by subtracting the mean and dividing by the standard deviation. These values of mean and standard deviation for the training set were used to normalise the test set.

A GUI was developed so that we could step through the price series of the test set and observe the movement in the latent space with a purple arrow as a marker. Figure 4.2 shows the training data, Figure 4.3 the test data and Figure 4.4 the latent space plot of means.

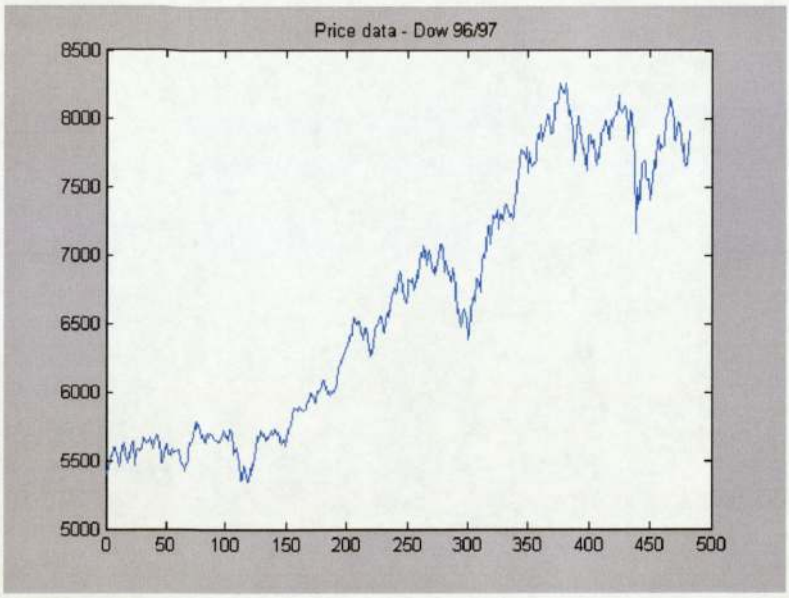


Figure 4.2: Daily Dow Jones Index data 1996-1997 used to train the model

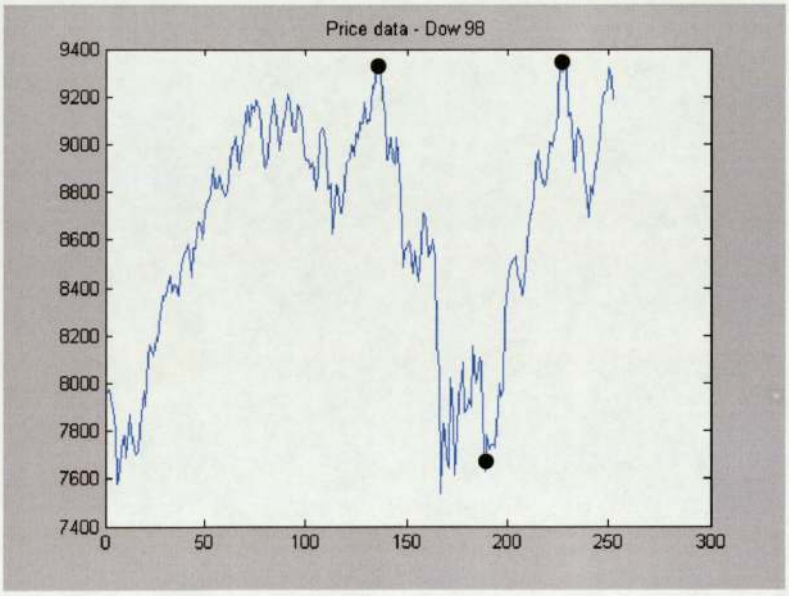


Figure 4.3: Daily Dow Jones Index 1998 used as test data, this is adjacent in time to the training data. Three key turning points highlighted with dots.

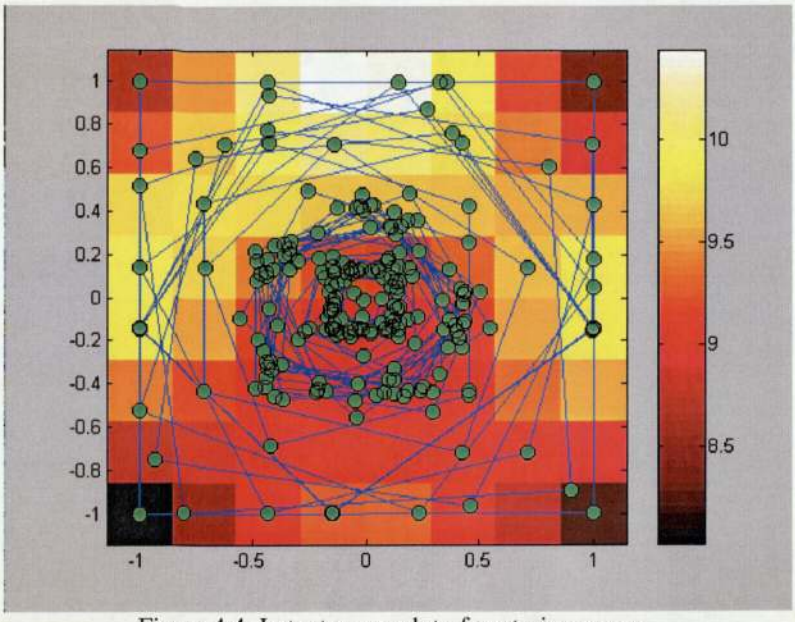


Figure 4.4: Latent space plot of posterior means.

Notice the 'circling' appearance of the latent space plot in Figure 4.4. The aim of this experiment was to investigate whether training the model on one dataset would enable us to detect regime shifts in a second dataset by studying aspects of the latent space plot. The three key turning points shown in Figure 4.3 were studied by stepping through the data points preceding the turning points and points immediately after. The movement in the latent space was observed to see if there were any patterns in the latent space corresponding to the regime shifts in the price series. So for example if a large jump in the latent space from one green circle to the next in the latent space was observed prior to each key turning point then this would be an encouraging sign. At this stage no conclusive results were achieved so further investigation was needed.

Our aim now was to try to use the GTM through time model to characterise the data sequences leading to regime shifts. In the rest of the chapter the following key experiments are discussed:

- Large jumps in latent space – looking at distances between adjacent means.
- Quantifying magnification factors using a data space distance calculation.
- The probability of data points given the model and outlier detection.

These three approaches were the most promising for predicting regime changes: a full evaluation is contained in Chapter 5. Further exploratory experiments were conducted with the aim of either refining the model or producing additional indicators that would increase the information we could extract from the visualisation plot. These experiments did not yield conclusive results but are briefly discussed:

- Price changes from latent squares.
- Price changes between different latent space regions.
- Directional curvatures to aid visualisation.
- Independent Component Analysis and use of a subspace matrix to pre-process data.
- Using trading volume as an input to the model and multi-market data.

4.3 Large jumps in latent space

As we move through a price series along the time axis a corresponding trajectory of means is traced out in the latent space plot. We can analyse the trajectory by looking at the distance between each pair of means corresponding to neighbouring points in the price series. Dow Jones Index 1998 daily data was used to train the model. We then calculated the distances between pairs of means in the latent space. Any distances greater than two standard deviations away from the mean were labelled as large and the corresponding points in the data series highlighted. Note that a test data set was not used in this experiment so there is a degree of hindsight in setting the threshold level. The aim of this experiment was to see if large jumps in the latent space corresponded to any significant regime changes in the data.

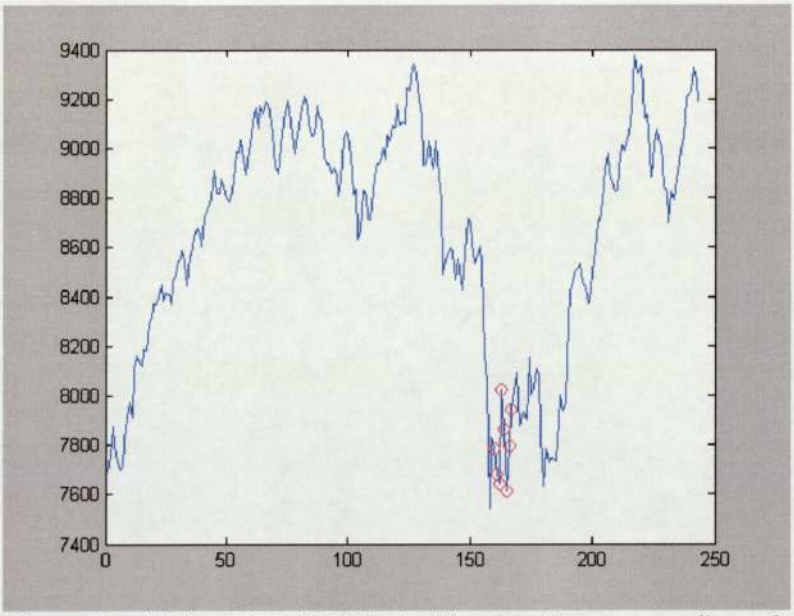


Figure 4.5: The price series with the points highlighted with red circles corresponding to large jumps in latent space.

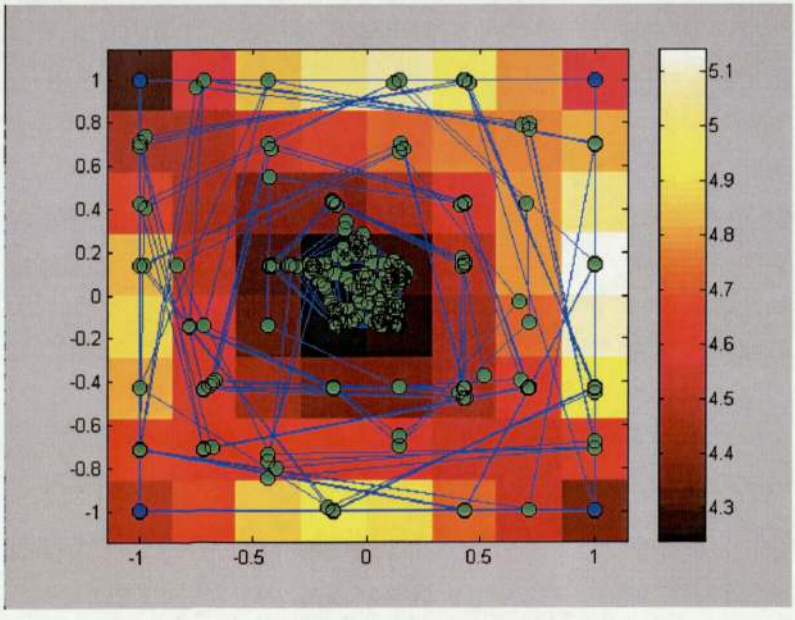


Figure 4.6: GTM through time latent plot. Blue circles show large jumps between latent points.

Figure 4.6 shows the latent space plot of posterior means. The large distance jumps are highlighted as blue circles. The corresponding points in the price series are highlighted as red circles in figure 5.1. The large jumps highlighted in the latent space occur as we move from one corner of the grid to another. The corresponding behaviour in the data series is a change from a downtrend to an uptrend so this is encouraging. However in this case we do not have prediction but are given a signal after the change has occurred.

4.4 Magnification factors – data space distance

Magnification factors measure the stretching of the manifold in data space. The magnification factors shown in the latent space plot are calculated at the latent points. In order to compute the total magnification factors between two points in the latent space an integral would need to be calculated. We can approximate this calculation by dividing the line between two points in the latent space into n equal segments. The $n+1$ points defining this partition are mapped to the data space and the straight-line distances between neighbouring data points are calculated. The values for all the segments are then summed to give a measure of distance between two points when mapped to the data space. This gives us a piecewise linear approximation to the desired value.

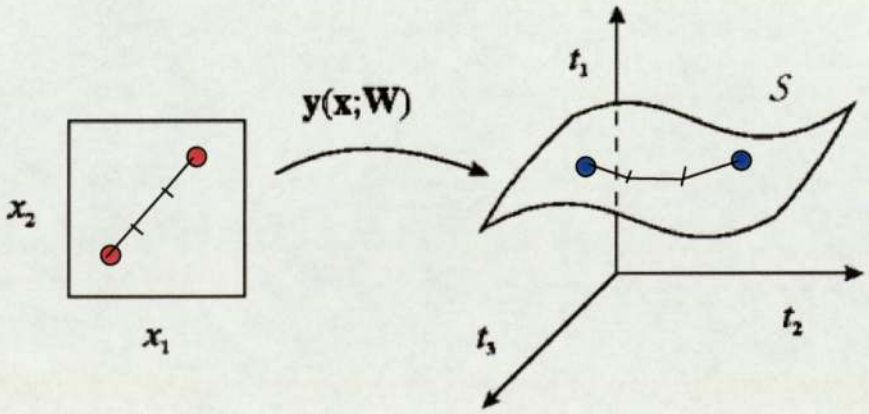


Figure 4.7: Magnification factors summed between two latent space means by mapping to data space and calculating a piecewise linear approximation.

The model was trained on Dow Jones daily data 1996-1997 and 1998 data was used for the test set as in Section 4.2. The results of the data space distance calculation for the test set are shown in Figure 4.8 below. The high values around the 150 mark correspond to a significant regime change in the price series, which is shown in Figure 4.9. On the basis of these results the magnification factor calculation looks like a possible indicator of regime changes.

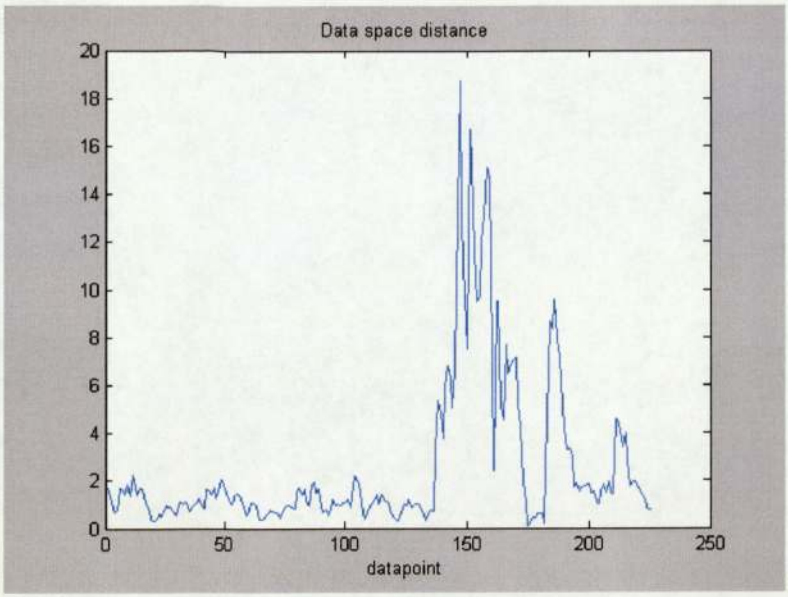


Figure 4.8: Data space distance calculation for Dow Jones Index 1998. The high values around the 150 mark correspond to a significant regime change in the price series.

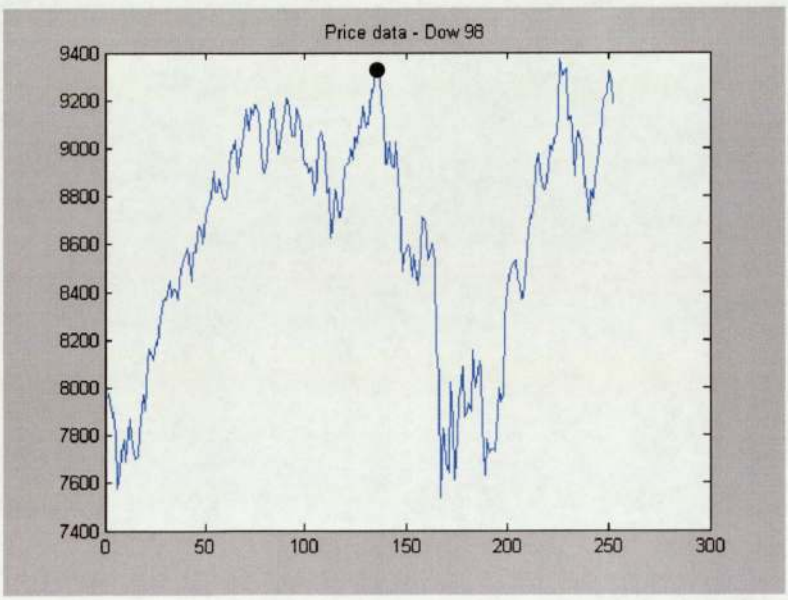


Figure 4.9: Dow Jones daily data 1998. The black dot highlights a regime change that corresponds to the high values around the 150 mark in figure 4.8.

4.5 Probability of data given the model

In this experiment the probability of the data given the model is calculated and displayed to analyse probabilities at key turning points in a price series. O_t is the observation at time t and Q_t is the state at time t. We want to compute the probability density of an observation at time t given the observations preceding it and after it and the model λ :

$$p(O_t \mid O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, \lambda)$$

$$= \sum_{i=1}^{K} p(O_t, Q_t = i \mid O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, \lambda) \text{ (where K is the number of states of the model)}$$

$$= \sum_{i=1}^{K} p(O_t, Q_t = i) p(Q_t = i \mid O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, \lambda) .$$

We have

$$p(Q_t = i \mid O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, \lambda) = \frac{p(O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, Q_t = i \mid \lambda)}{\sum_{j=1}^{K} p(O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, Q_t = j \mid \lambda)} .$$

But $p(O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, Q_t = i \mid \lambda) = \sum_{j=1}^{K} \alpha_{t-1}(j) a_{ji} \beta_t(i)$ where $a_{ji} = p(Q_t = j \mid Q_{t-1} = i)$. The variables α, β are defined in the Appendix.

So finally

$$p(O_t \mid O_{1,\ldots,t-1}, O_{t+1,\ldots,T}, \lambda) = \sum_{i=1}^{K} p(O_t \mid Q_t = i) \left[\frac{\sum_{j=1}^{K} \alpha_{t-1}(j) a_{ji} \beta_t(i)}{\sum_{i=1}^{K} \sum_{j=1}^{K} \alpha_{t-1}(j) a_{ji} \beta_t(i)} \right] .$$

The model was trained as in section 4.2 using 1996-1997 daily Dow Jones Index data. Figure 4.11 shows a graph of the log of the data probability given the model for the test set, which is Dow Jones Index daily data 1998 shown in figure 4.10. The low probability region at around the 150 mark corresponds to a significant regime change in the price series so this suggests that this probability calculation could possibly be used for prediction of regime changes.

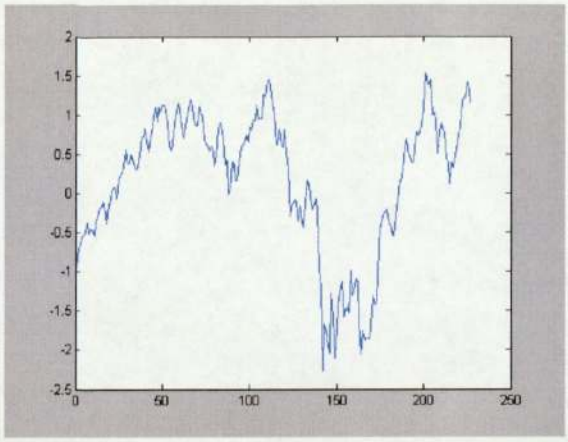


Figure 4.10: Dow Jones Index daily 1998. Notice the reversal around the 150 mark.

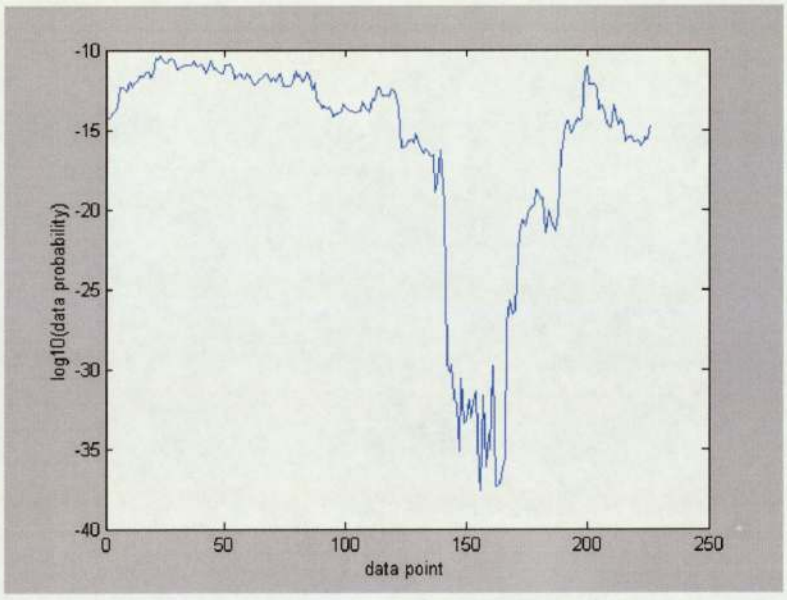


Figure 4.11: Log of probability of data given the model.

The calculation performed cannot be used as a prediction indicator as it uses the entire data set but is useful for analysis. A modified version of the calculation using only the forward variable alpha was derived for use as an indicator: we calculate $p(O_t \mid O_{1,\ldots,t-1}, \lambda)$, that is the probability of an observation at time t given the previous observations and the model.

4.6 Outlier detection

Outliers are observations that are not well fitted by the model. Whenever data levels are too high or too low compared to neighbouring points we call such points outliers. In the context of this project we investigated whether outilier points corresponded to regime changes.

An outlier point was deliberately created for Dollar Index daily data. The graph of the data can be seen in figure 4.12 with the outlier point at the 200 mark. We then trained the model on this data and used the data probability calculation to investigate whether the probability for this point would significantly vary from values for other points. The data probability graph in figure 4.13 shows that the probability for the outlier point is significantly lower than for other points in the data set. These results suggest that an abrupt change in a market may possibly be detected by the data probability calculation, although in this case the abrupt change was artificially created. We have a small window of points with low probability values rather than just one point due to the fact that delay vectors are being used.

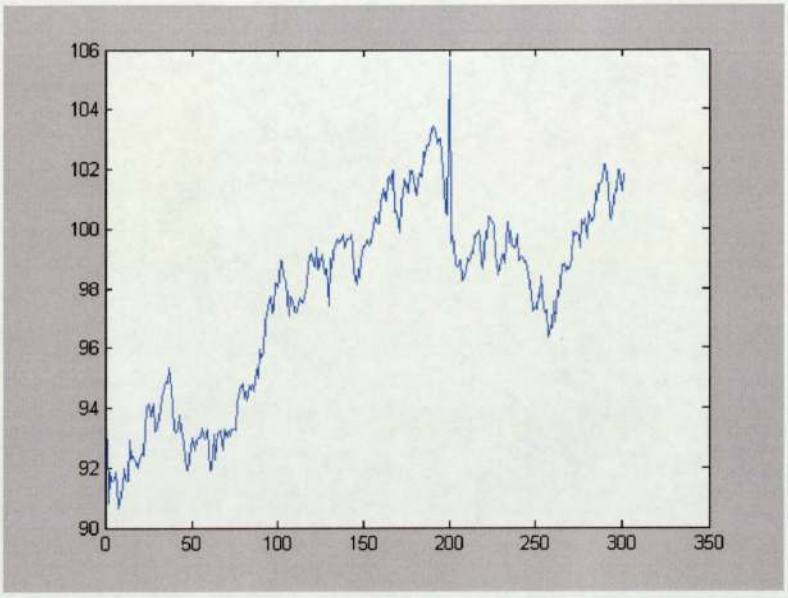


Figure 4.12: Dollar Index daily data. Outlier point at t = 200.

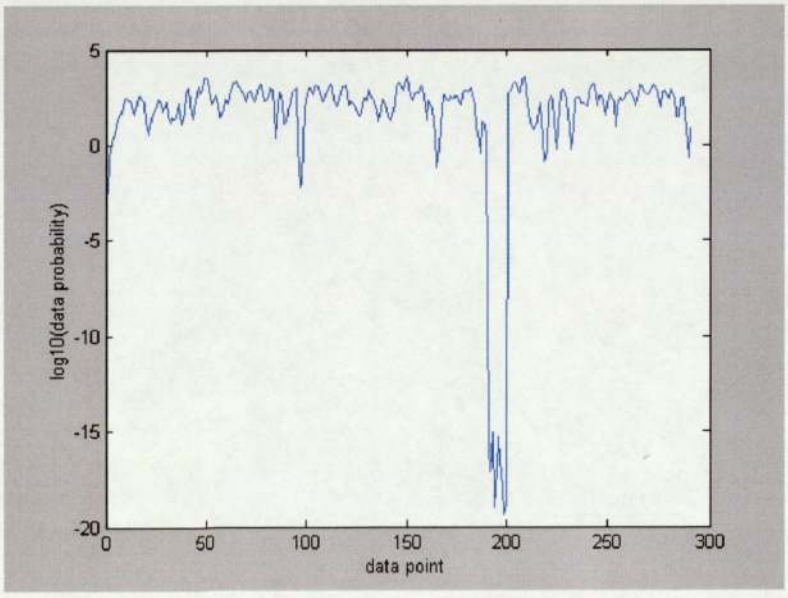


Figure 4.13: Probability of data given the model.

4.7 Further Experiments

The following experiments show some of the different avenues that were explored during the research project but did not yield conclusive results that could be used.

4.7.1 Price changes from latent squares

The aim of this experiment was to analyse price changes as a function of squares in the latent space grid. The experiment was performed using the Dow Jones daily data 1996-1997 for training and 1998 data as the test set. We first established the posterior means corresponding to each latent square in the grid i.e. which green circles lie in each latent square. Then for each green circle the next green circle in the trajectory was established. The corresponding absolute price change in the financial data was calculated. For each latent square the price changes for each green circle within it are summed. This then gives a value for each latent square that is displayed using a coloured grid as for the magnification factors. The results can be seen in Figures 4.14 and 4.15 below. The two squares with the highest price changes, shaded yellow and white in Figure 4.14, correspond to a significant turning point in the price data which is highlighted with a purple circle in Figure 4.15.

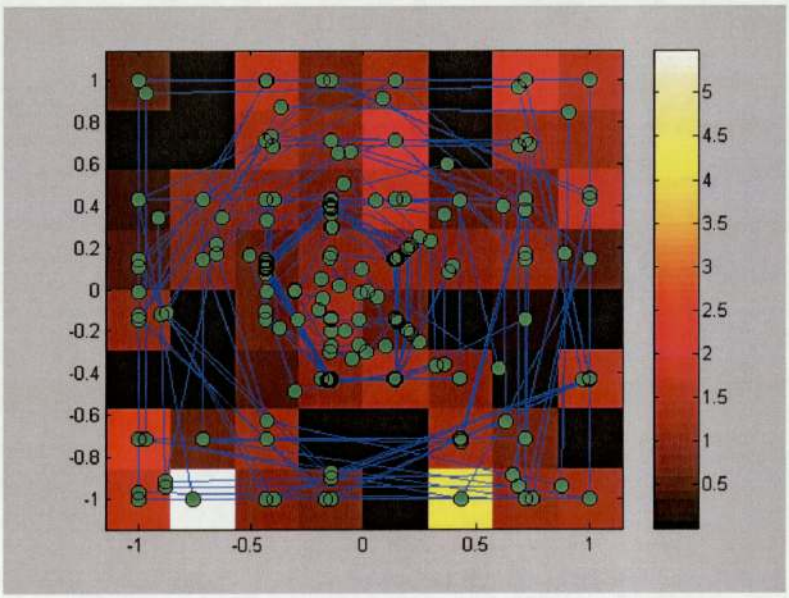


Figure 4.14: Price changes from each latent square. The yellow square in the grid corresponds to the point highlighted in the Figure 4.15.

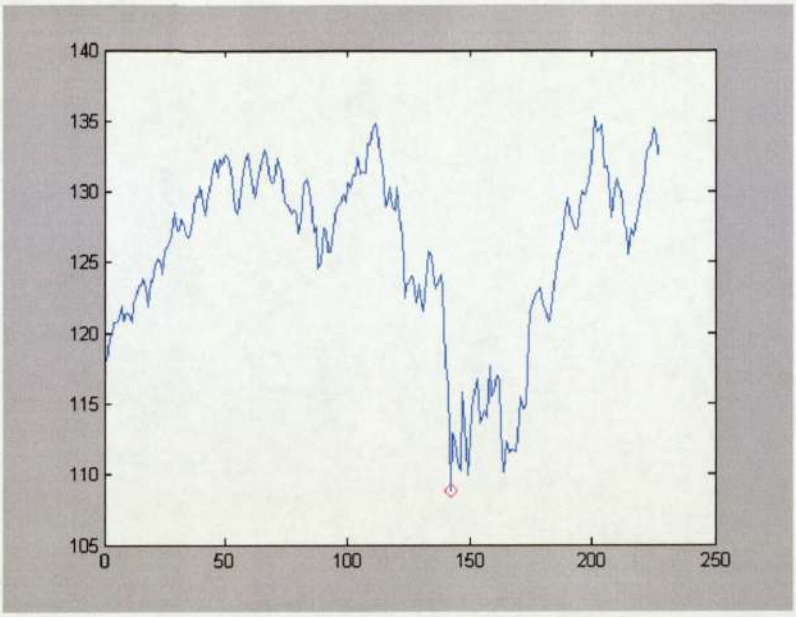


Figure 4.15: Dow Jones Index 1998. Purple circle corresponds to latent point in yellow square in latent plot Figure 4.14.

This experiment was performed as an investigation into price changes from latent space squares. In order to use this for prediction of regime changes a latent grid based on the training set rather than the test set – which uses perfect hindsight – would need to be used. Since the structure of the training data is often considerably different to the test data and the latent space trajectory is different, a latent grid based on price changes in the training set would be of little prediction value.

4.7.2 Price changes between regions

The magnification factors in a typical GTM through time plot in this project are smaller in the centre of the plot and greater around the perimeter as in Figure 4.6. Also the trajectory of means has a 'circling' appearance with movement from the centre of the plot to the outer region. In this experiment we attempted to quantify this behaviour by splitting the latent space into four regions as shown in Figure 4.16. The inner four squares are region one. The outer squares bordering this region are region two. The outer squares bordering region two are region three. The outer squares bordering region three are region four. Average price changes between different regions were calculated. The experiment was performed using Dow Jones Index 1996-1997 daily data for training and Dow Jones Index 1998 daily data as the test set.

From the top left of the grid in Figure 4.17 the regions are represented by one to four left to right, and one to four moving vertically down. So for example the first square in the top left of the grid represents average price changes from region one to region one. The highest average price changes are from region three to region two represented by the white square. These results did not enable conclusive arguments to be made about changes between regions of the latent space defined in this way and corresponding price changes and so the calculation was not used in subsequent experiments.

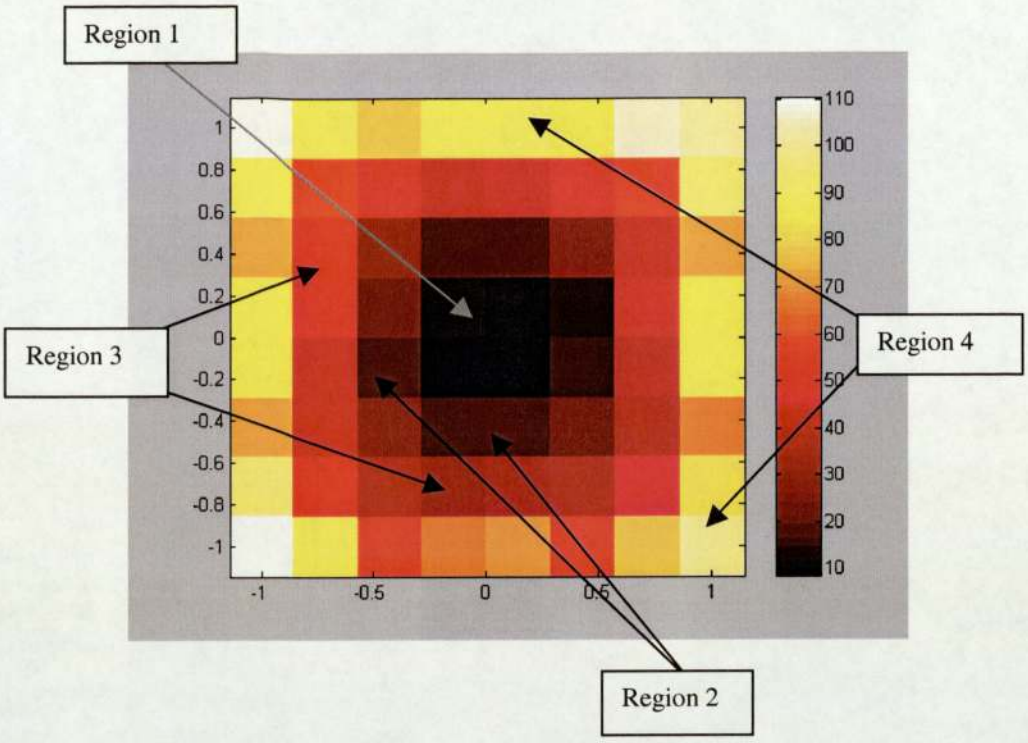


Figure 4.16: Latent space grid split into four regions.

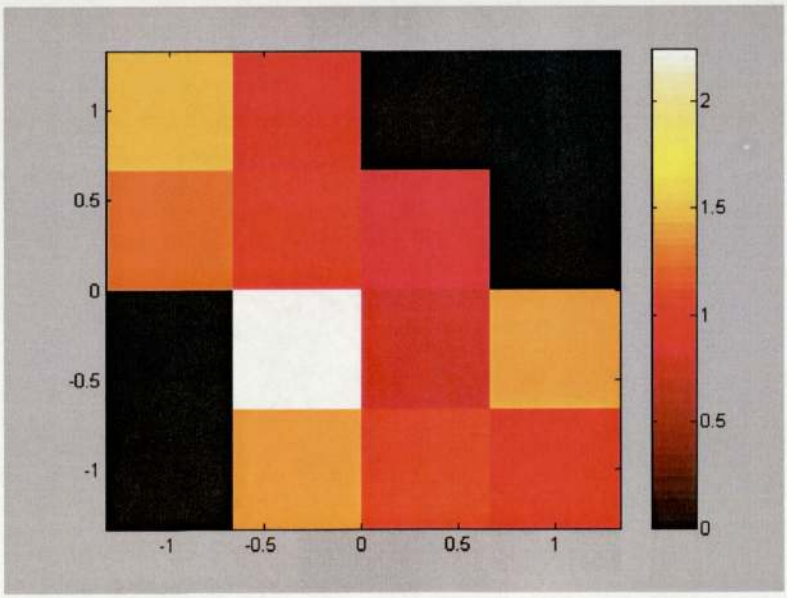


Figure 4.17: Average price changes between regions of the latent space.

4.7.3 Directional Curvatures

When injecting a two-dimensional sheet into a high dimensional data space, the projection manifold may form complicated folds that cannot be measured by using magnification factors alone; instead we can use directional curvatures (Tino, Nabney and Sun, 2001) to measure the amount of folding. Using Dow Jones 1996-1997 daily data for training and 1998 data as the test set, the directional curvatures are displayed in Figure 4.19. The posterior means are not displayed. The directional curvatures are reasonably close to 1; this means that the manifold is reasonably flat and therefore represents the data well.

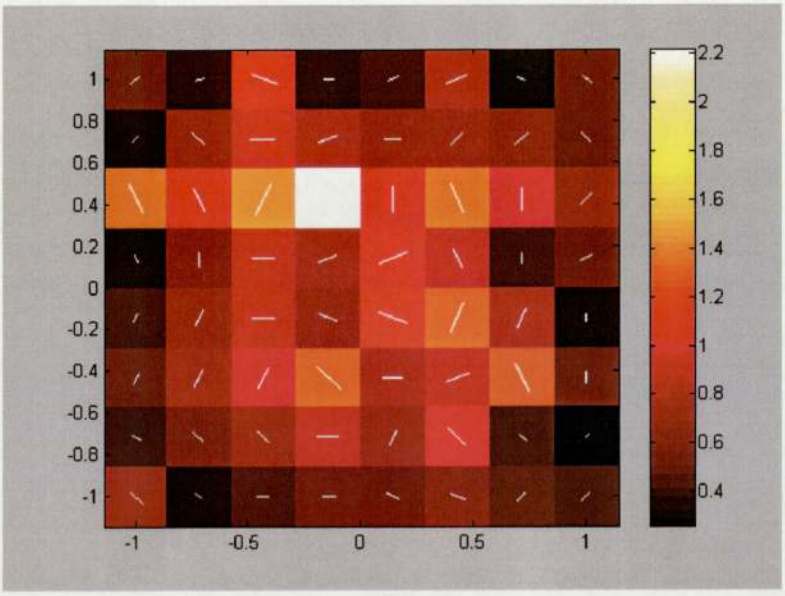


Figure 4.19 Directional curvature plot (test data set). For each latent space centre the length of the white direction line and the degree of shading are proportional to the directional curvature.

4.7.4 Independent Component Analysis

Independent Component Analysis (Haykin, 1998) is used to separate a source matrix into statistically independent components. These components do not have to be orthogonal. For this experiment a delay vector matrix of column dimension 25 was formed using Dollar-Yen daily data. This delay vector matrix was then processed using the FastICA software program to perform Independent Component Analysis on this matrix. The original matrix of column dimension 25 was transformed into a matrix of dimension six, with each column being as statistically independent as possible. This new matrix was input into the model as the training data. The reason for doing this is to pre-process the data so that in some sense the underlying dynamics of the system are more accurately captured. The latent space plot that is produced is lacking in structure and so this method was not used in any further experiments. The data used is shown in Figure 4.20 and the latent plot in Figure 4.21.

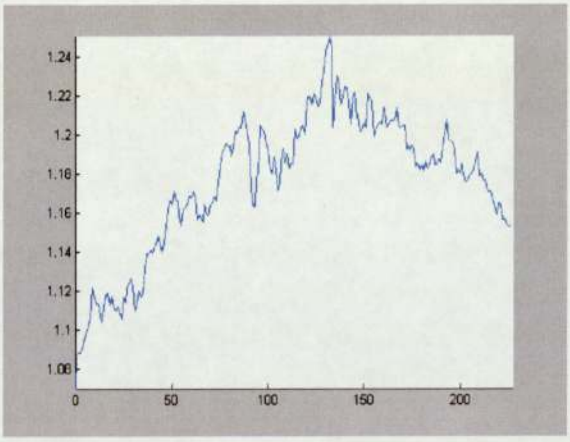


Figure 4.20: Daily Dollar-Yen data used to train the model

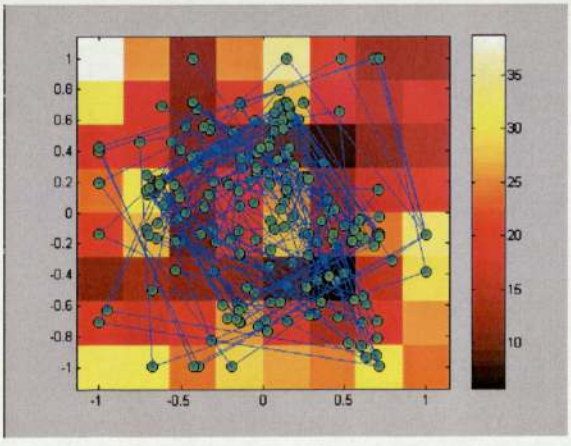


Figure 4.21: Latent space plot. Data pre-processed using ICA.

4.7.5 Subspace matrix

The data used in this experiment was Dow Jones Index daily data from 1983-1984. A delay vector matrix of column dimension 50 was formed. Using singular value decomposition this matrix can be expressed in the form $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. By analysing the spectrum of the delay vector matrix a value for a reduced dimension matrix which would still retain the information of the original matrix was deduced. The aim of this was to capture the dynamics of the underlying system more accurately and therefore improve the model. The justification for doing this is embedding theory which is discussed in Section 3.22. So we have

$$\boldsymbol{A}_1 = \boldsymbol{U}_1\boldsymbol{\Sigma}_1$$

where $\boldsymbol{U}_1$ contains the first d columns of $\boldsymbol{U}$, and $\boldsymbol{\Sigma}_1$ contains the first d rows and first d columns of $\boldsymbol{\Sigma}$ .The matrix **V** is not included in the calculation of the subspace matrix as it is only a rotation matrix. For the data used, the subspace dimension was $d = 20$. The model was trained on the subspace matrix but the latent plot produced lacks structure with trajectory of points oscillating from one side of the latent space to the another so this method was not used in further experiments. The training data is shown in Figure 4.22 and the latent space plot is shown in Figure 4.23.

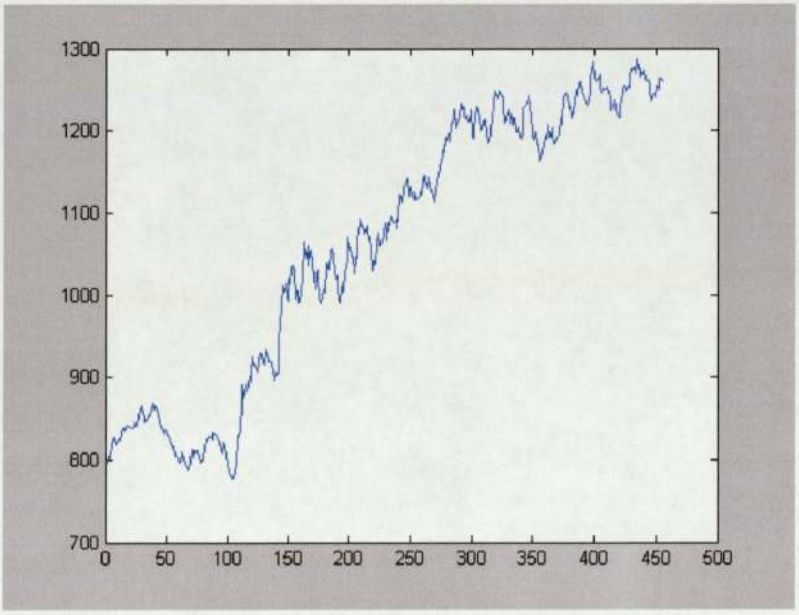


Figure 4.22: Dow Jones Index daily data 1983-1984

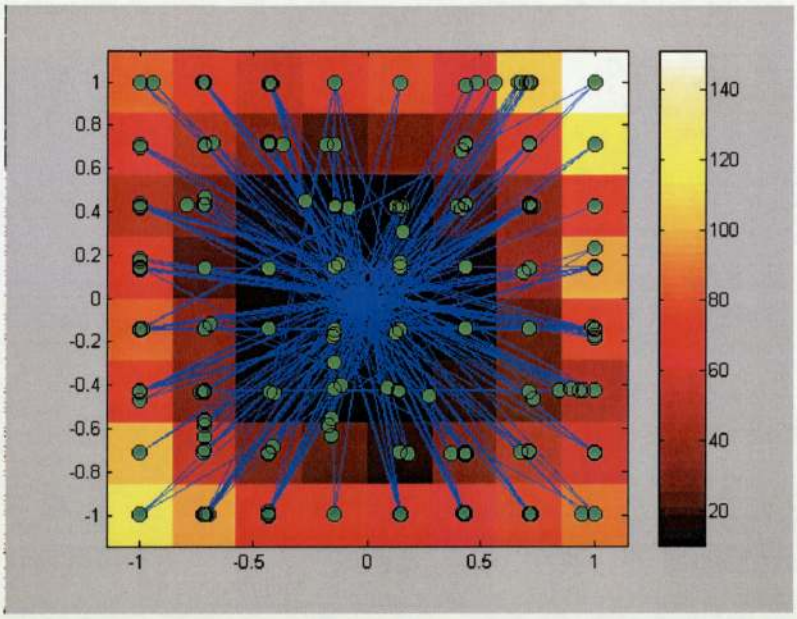


Figure 4.23 Latent space plot of means. The trajectory produced lacks structure with the means jumping from one side of the plot to another.

4.7.6 Using trading volume as an input

In this experiment the model was trained using both price and trading volume information. Daily Dollar/Yen data for 1988 to 1989 was used to train the model and 1990 data was used for testing. The test price data is shown in figure 4.24. The latent space plot produced was lacking structure with the trajectory of means jumping from one part of the latent space to another in an erratic fashion. This is shown in figure 4.25. This avenue was not pursued further as adding volume information added no further insight.

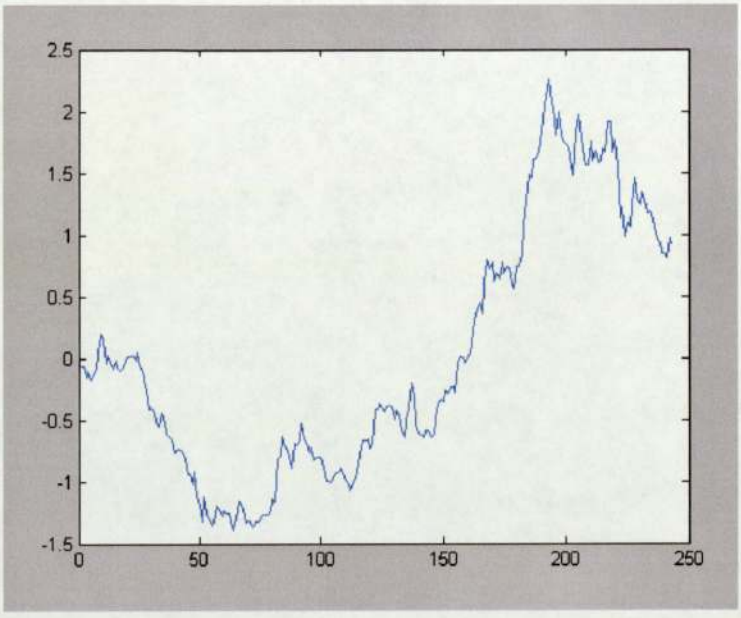


Figure 4.24: Normalised Dollar/Yen daily data 1990.

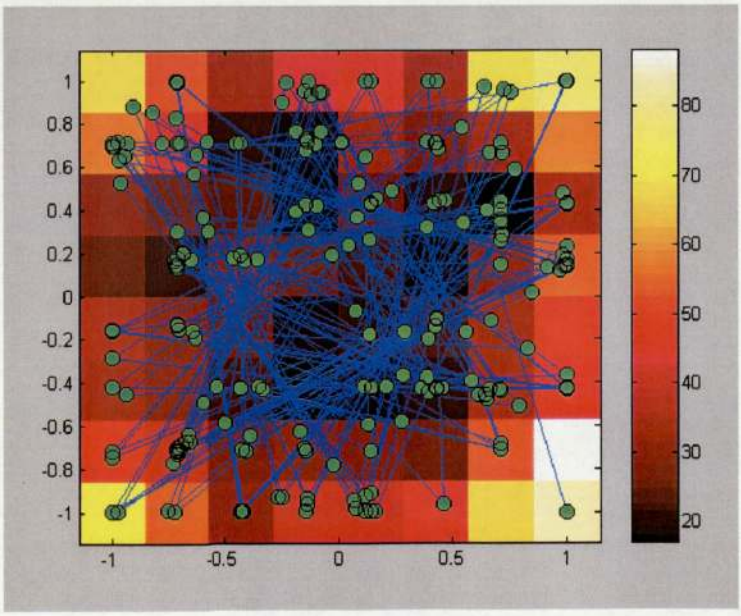


Figure 4.25: Plot of posterior means

4.7.7 Multi-market data

The aim of this experiment was to investigate training the model on multi-market data. Three currency indices were used to train the model. Daily Dollar Index, Dollar/Yen and Dollar/Pound were pre-processed using price difference and combined to form a set of delay vectors. The Dollar Index is the US dollar weighted against a basket of currencies. Data from 1997 to 1998 was used to train the model and 1999 data was used for testing. The data used (normalised to have zero mean and standard deviation one) is shown in Figure 4.26 and the latent space plot is shown in Figure 4.27.

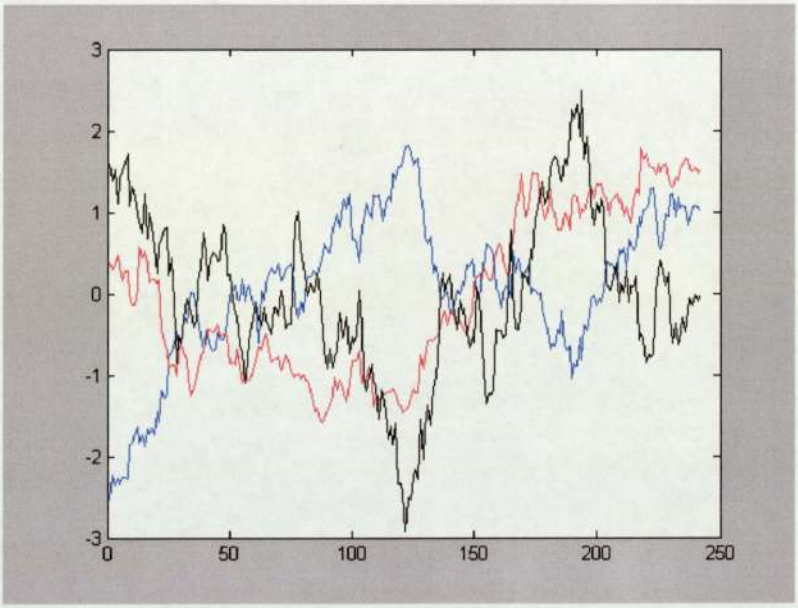


Figure 4.26: Normalised currency data. Blue – Dollar Index, Red – Dollar/Yen, Black – Pound.

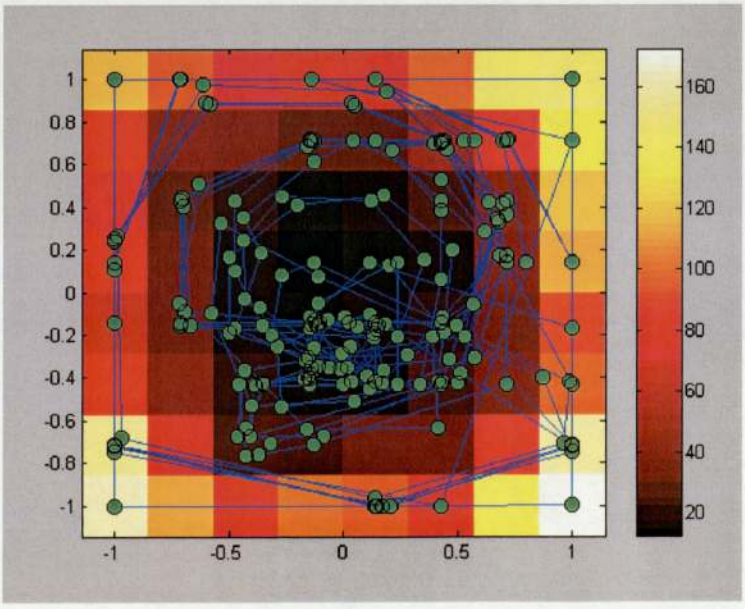


Figure 4.27: Latent space plot for multi-currency experiment.

Stepping through the data set and observing movement in the latent space we were unable to draw useful conclusions from this experiment. Further investigation is needed to correlate a change in the latent space to a regime, as there are three different markets.

4.8 Summary

In this chapter we first considered generalisation of the GTM through time model using a test data set having trained the model. We found that the data needed to be pre-processed using price differences to enable generalisation. This meant that a characteristic plot was produced in the latent space that had a 'circling' trajectory. We then covered key experiments that produced indicators that may possibly be of use in predicting regime changes in financial data: large jumps in the latent space, changes in magnification factors (quantified using a data space distance calculation) and the probability of the data given the model. Further experiments that did not provide conclusive results but showed avenues of research were also discussed.

Chapter 5

Financial Time Series -Analysis of Regime Shifts

5.1 Introduction

In Chapter four we covered key experiments that showed indicators which could possibly be used for prediction of regime shifts. In order to establish statistically significant results the indicators need to be tested on a large number of data points. Also our definition of a regime shift has been somewhat qualitative and needs to be more precise. In order to simplify the problem of defining regime shifts we will restrict ourselves to relative highs and lows in the financial data. In this chapter we will investigate the use of two key indicators for prediction of regime shifts.

5.2 Quantification of Regime changes and comparison with model predictions

An effective method was found of calculating relative highs and lows. If a point is greater than all n points before it and all n points after it then it is a relative high. If a point is less than all n points before it and all n points after it then it is a relative low. Heuristically n was set to 10. A relative low is a point where a market in a downtrend changes into an uptrend as shown in Figure 1.1. A relative high is a point where a market in an uptrend changes into a downtrend as shown in Figure 1.2. Our restriction to highs and lows means we miss other types of regime shifts for example a sideways market changing into an uptrend or downtrend. The high-low method was used to mark up data sets so the key turning points were flagged. Figure 5.1 shows marked up Dollar/Yen daily data. The points highlighted with red circles are the relative highs and lows.

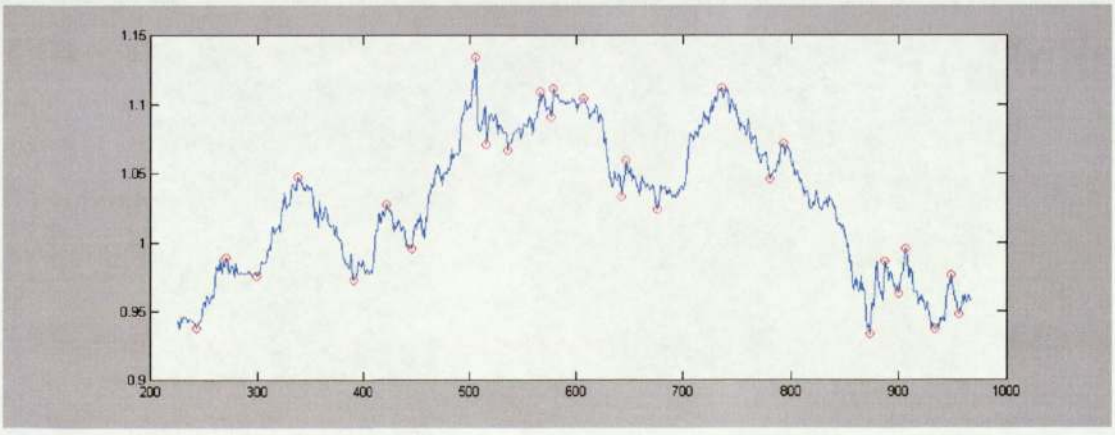


Figure 5.1: Dollar/Yen daily data. Red circles show automated method of selecting highs and lows.

In Chapter four three key indicators were mentioned: large jumps in latent space, magnification factors and data probability. Further investigation showed that large latent space jumps and magnification factors produced the same signals. So only two of the key indicators discussed in Chapter four were used to attempt to predict regime changes. Firstly magnification factors calculated using data space distance are considered. Secondly the probability of the data given the model is considered. In order to use these two methods as indicators we calculated a set of distances and probability values for the training set. We then calculated the mean of the distances and added n standard deviations to obtain a threshold value. We calculated logarithms of the probabilities and subtracted n standard deviations to obtain a probability threshold value. The distance and probability calculations were then carried out for our test set. Points that exceeded our threshold values were highlighted.

Several data sets were used and these are shown in Table 5.1. For the daily data, two years of data was used for training and one year for the test set. For the intraday data 500 points were used for training and 250 points for testing. A sliding window was used so that after the first train/test cycle the window was moved forward one year for daily data and 250 points for intraday data and the train/test cycle was repeated.

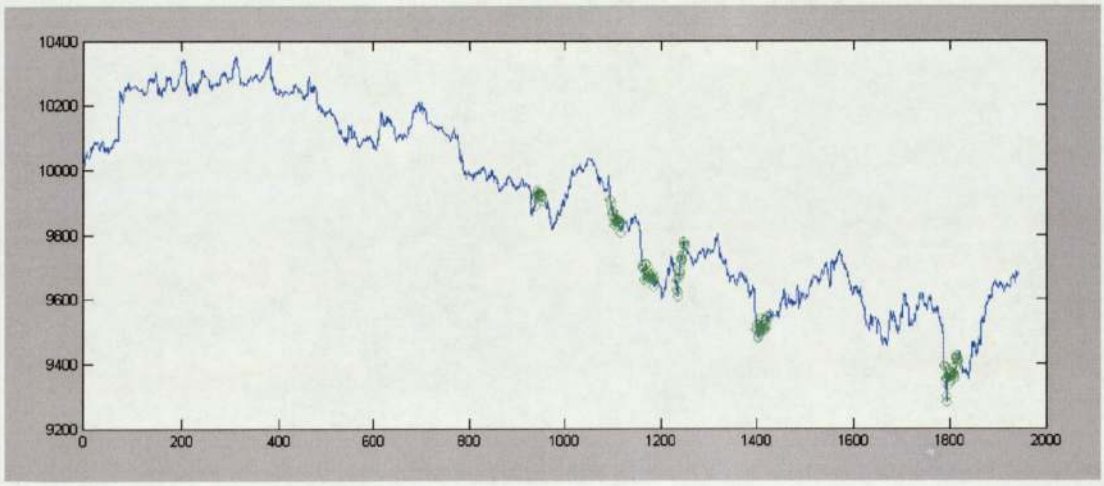


Figure 5.2: Dow Jones Intraday data 5-min. Green circles show low probability points.

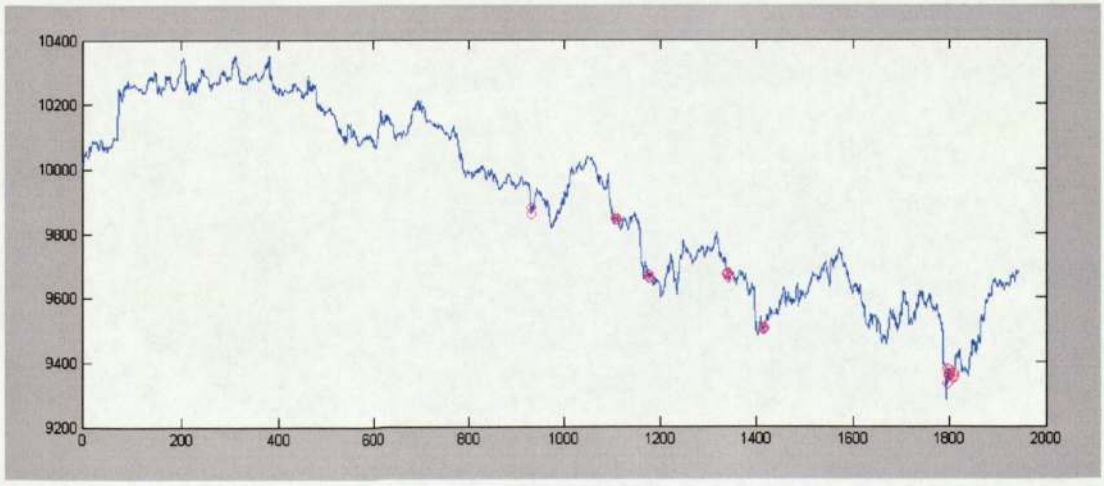


Figure 5.3: Dow Jones Intraday data 5-min. Purple circles show high distance points.

Market	Number of data points
Dollar/Yen daily 1986-1999	3534
Dow Jones Index daily 1981-1987	1770
Intel daily 1990-1999	2526
Dow Jones Index 5 minute	1942

Table 5.1: Financial data used in experiments

In figure 5.2 we can see approximately one month of Dow Jones intraday data (5-minute bars). The green circles show the points detected as having a low probability using a threshold value of 2 standard deviations from the mean. In Figure 5.3 we have the same data as in Figure 5.2 with purple circles showing the points detected by the high data space distance calculation again using a threshold value of 2 standard deviations from the mean. In both figures there are clusters of points detected using our current indicators. So in order to produce signals that we can compare with the marked up high and low points we need to refine our indicators. This can be done by taking the first point in a cluster as a signal of an expected regime shift. Figure 5.4 shows the signals produced by this method. The green circles show the low probability signals and the purple circles show the magnification factor (high data space distance) signals.

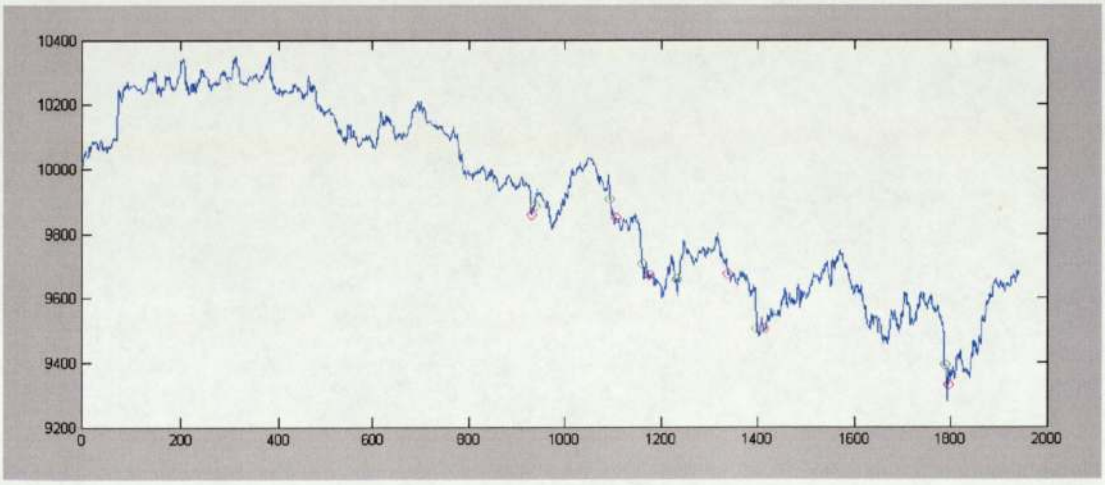


Figure 5.4: Dow Jones Intraday data 5-min. Purple and green circles show high distance and low probability signals respectively.

The low probability and high data space distance regime change signals were calculated for the four data sets shown in Table 5.1. A signal was classified as valid if it was between 10 and 0 points preceding a marked high or low data point. This means that if a signal occurred at the same time as a marked data point it was still classified as a valid prediction although strictly speaking it was a detection and did not give early warning of a regime change. The results of the evaluation are shown in table 5.2.

Market: Dow Jones Index intraday 5 minute Number of data points: 1942 Number of marked points: 106					
Threshold (stdev.)	**1.0**	**1.5**	**2.0**	**2.5**	**3.0**
Low probability signals	12	8	6	6	5
Correct low prob. signals	10	7	6	4	2
High distance signals	16	12	6	5	4
Correct high dist. signals	11	7	4	2	2
Market: Dollar/Yen daily 1986-1999 Number of data points: 3534 Number of marked points: 158					
Low probability signals	21	21	15	11	4
Correct low prob. signals	11	8	9	9	4
High distance signals	32	31	24	19	14
Correct high dist. signals	17	16	12	10	7
Market: Dow Jones Index daily 1981-1987 Number of data points: 1770 Number of marked points: 111					
Low probability signals	8	5	5	3	2
Correct low prob. signals	4	3	2	2	0
High distance signals	23	21	12	10	2
Correct high dist. signals	13	13	7	5	0
Market: Intel daily 1990-1999 Number of data points: 2526 Number of marked points: 144					
Low probability signals	15	13	9	6	4
Correct low prob. signals	7	7	2	1	1
High distance signals	28	25	17	9	6
Correct high dist. signals	17	13	10	4	3

Table 5.2: Results of regime change prediction on different data sets

The threshold levels were varied between one and three standard deviations of the training set values. We can see that as the number of standard deviations increases the number of signals decreases as we would expect. To clarify the results above let us consider the Dow Jones intraday data. There are a total of 1942 data points and 106 points marked as either a high or a low. In the column with standard deviation equal to one the probability indicator produced 12 signals of an expected regime shift and 10 of these signals were between 10 and 0 points preceding a marked point and therefore classified as correct. The high distance signals are listed similarly.

One problem is that a regime change that is not a high or low may be classified as 'incorrect' using the current method. It seems encouraging that a significant percentage of the signals produced are classified as correct. However given the numbers of marked points and the much fewer numbers of signals produced the indicators are missing most of the regime changes that we wish to predict.

5.3 Summary

In this chapter we first quantified the definition of a regime shift restricting the type of change we are looking to predict as corresponding to a relative high or relative low. An automated method of marking up data was discussed. Two key indicators were then used to predict regime changes on a number of different data sets. The predictions were compared with the marked up points and the results discussed. It was found that a significant percentage of the predicted regime changes are correct. However most of the regime changes in the data are missed so the indicators are not satisfactory for trading purposes.

Chapter 6

Conclusion

The aim of the project was to study the use of the GTM through time model for predicting regime changes in financial market data. We covered the theory behind GTM and GTM through time and the practical implementation of the model. Visualisation of the data in the latent space was a key element of this project. However it has not been possible to correlate movement between one region of the latent space to another with regime changes in the financial data. Some success was achieved by calculating changes in magnification factors between two means in the latent space and using this as an indicator of regime changes. Also the probability of the data given the model as an indicator showed promising results. However the indicators failed to capture most of the regime changes when tested on large data sets and so could not be used for trading purposes.

Further work that could be carried out is listed below:

- Refinement of current indicators and development of new ones.
- Model itself could be refined so that it is does not use unsupervised learning.
- More work to attempt to correlate movement in the latent space with regime changes.
- Improve the automated method of marking up data to include other types of regime shifts.
- Test model on short time frames using more intraday data.
- Test model on data with randomised order to break the temporal structure of the data; therefore anything that we can see in normal data that we can not see in the randomised data should represent genuine structure.
- Change the feature space; alter the input of data using delay vectors to another method.

One of the aims of the project was to investigate the use of GTM through time for developing a trading system. With the current indicators there are too few signals for the model to be a tradable system and a mathematically simple breakout system with good money management principles would produce a higher frequency of trades and better results. This conclusion was reached after discussions with a former derivatives trader who is currently involved in building automated trading systems. It is also possible that this model is not appropriate for studying financial data but further investigation would be needed to confirm this.

Appendix: Hidden Markov Models

A Hidden Markov Model is a finite set of *states*, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called *transition probabilities*. In a particular state an outcome or *observation* can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are 'hidden' to the outside; hence the name Hidden Markov Model (Dugad and Desai, 1996).

In order to define a HMM completely, following elements are needed:

- The number of states of the model, N.
- The number of observation symbols in the alphabet, M. If the observations are continuous then M is infinite.
- A set of state transition probabilities $A = \{a_{ij}\}$

$$a_{ij} = p\{Q_{t+1} = j \mid Q_t = i\} \qquad 1 \le i, j \le N$$

where Q_t denotes the current state.

Transition probabilities must satisfy the following constraints

$$a_{ij} \ge 0 \qquad 1 \le i, j \le N$$

and

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad 1 \le i \le N .$$

- A probability distribution in each of the states $B = \{b_j(o_k)\}$

$$b_j(o_k) = p(O_t = o_k \mid Q_t = j\} \qquad 1 \le j \le N, 1 \le k \le M$$

where o_k denotes the k^{th} observation symbol in the alphabet, and O_t the observation at time t.

The following constraints must be satisfied:

$$b_j(o_k) \ge 0 \qquad 1 \le j \le N, 1 \le k \le M$$

and

$$\sum_{k=1}^{M} b_j(o_k) = 1 \qquad 1 \le j \le N .$$

If the observations are continuous then we will have to use a continuous probability density function, instead of a set of discrete probabilities. In this case we specify the parameters of the probability density function. Usually the probability density is approximated by a weighted sum of M Gaussian distributions Ψ,

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \Psi(\mu_{jm}, \Sigma_{jm}, o_t)$$

where,

c_{jm} = weighting coefficients

μ_{jm} = mean vectors

Σ_{jm} = Covariance matrices

c_{jm} satisfies the constraints

$$c_{jm} \geq 0, \qquad 1 \leq j \leq N, \qquad 1 \leq m \leq M$$

and $$\sum_{m=1}^{M} c_{jm} = 1, \qquad 1 \leq j \leq N.$$

- The initial state distribution $\pi = \{\pi_i\}$ where

$$\pi_i = p\{Q_1 = i\}, \qquad 1 \leq i \leq N.$$

Therefore we can use the compact notation $\lambda = (A, B, \pi)$ to denote a HMM.

The Forward-Backward algorithm

Consider a model defined by $\lambda = (A, B, \pi)$. We wish to find $\lambda^* = \underset{\lambda}{\operatorname{argmax}}\, p(O \mid \lambda)$ i.e. to find model parameters such that the probability of the observation sequence given the model is maximised. This can be done using the Baum-Welch algorithm.

The forward procedure

We define $\alpha_i(t) = p(O_1 = o_1, ..., O_t = o_t, Q_t = i \mid \lambda)$ which is the probability of seeing the partial sequence $o_1, ..., o_t$ and ending up in state *i* at time t. We can define $\alpha_i(t)$ recursively as:

1. $\alpha_i(1) = \pi_i b_i(o_1)$
2. $\alpha_j(t+1) = \left[\sum_{i=1}^{N} \alpha_i(t) a_{ij}\right] b_j(o_{t+1})$
3. $p(O \mid \lambda) = \sum_{i=1}^{N} \alpha_i(T)$ (where T is the final time point).

The backward procedure

The backward procedure is similar: $\beta_i(t) = p(O_{t+1} = o_{t+1},....,O_T = o_T \mid Q_t = i,\lambda)$

This is the probability of the ending partial sequence $o_{t+1},.....,o_T$ given that we started at state *i* at time *t*. We can efficiently define $\beta_i(t)$ as:

1. $\beta_i(T) = 1$
2. $\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_j(t+1)$
3. $p(O \mid \lambda) = \sum_{i=1}^{N} \beta_i(1) \pi_i b_i(o_1)$

We now define $\gamma_i(t) = p(Q_t = i \mid O,\lambda)$ which is the probability of being in state *i* at time *t* for the state sequence *O*.

It can be shown that $\gamma_i(t) = \dfrac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$.

We also define $\xi_{ij}(t) = p(Q_t = i, Q_{t+1} = j \mid O,\lambda)$ which is the probability of being in state *i* at time *t* and being in state *j* at time *t+1*.

It can be shown that $\xi_{ij}(t) = \dfrac{\gamma_i(t) a_{ij} b_j(o_{t+1}) \beta_j(t+1)}{\beta_i(t)}$.

The above explanation is an outline of the main points. For a detailed account see Rabiner, 1989.

Bibliography

C.M. Bishop, G.E. Hinton, and I.G.D. Strachan. *GTM through time*. IEE International Conference on Artificial Neural Networks, 1997.

C. M Bishop, M. Svensen, and C. K. I. Williams. *GTM: The Generative Topographic Mapping*. Neural Computation, 1997.

C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

R. Dugad and U.B. Desai. *A tutorial on hidden markov models*. Technical Report SPANN-96.1, Indian Institute of Technology, 1996.

Lawrence R. Rabiner. *A tutorial on hidden markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2): 257-284, February 1989.

P. Tino, I. Nabney and Yi Sun. *Using Directional Curvatures to Visualize Folding Patterns of the GTM Projection Manifolds*. In Artificial Neural Networks – ICANN, 2001.

I. Nabney. *Netlab Algorithms for Pattern Recognition*. Springer 2001.

L. Noakes. *The Takens Embedding Theorem*. International Journal of Bifurcation and Chaos 1, 867-872, 1991.

S. Haykin. *Neural Networks A Comprehensive Foundation*. Prentice Hall 1999.

T. Kohonen. *Self-organized formation of topologically correct feature maps*. Biological Cybernetics, 43:59--69, 1982.

J. Murphy. *Technical Analysis of the Financial Markets*. New York Institute of Finance 1999.

Tradestation 2000i. Omega Research.

FastICA. Laboratory of Information and Computer Science, Helsinki University of Technology.

Ben Goertzel, IntelliGenesis Corp, 1998.
http://www.goertzel.org/dynapsyc/1999/NonlinearTimeSeriesAnalysis.html