Lecture 9

Time series prediction
Prediction is about function fitting

To predict we need to **model**

There are a bewildering number of models for data – we look at some of the major approaches in this lecture – but this is far from complete

We'll start by looking at the difference between **function** and **curve** modelling approaches
A forecast, $y[t]$, is estimated from a function of observables $x$. The latter are not necessarily even from the past of $y[t]$. Critically, $x$ is not just the time variable.

Infer a manifold or response curve through retrospective data for use on new samples.
Example – rolling regression

\(x\) corresponds to a set of past samples, say \(x = (y[t-L],...,y[t-L+h])\). We develop a mapping from \(x\) to \(y[t]\) over a training set then use this mapping for subsequent forecasting.

The samples in \(x\) don't even need to come from \(y\) though – so this is really flexible.
What form could the mapping take?

Our mapping function can be any universal approximation approach – this will include (but not limited by), for example:

- Gaussian processes
- Neural networks
- Basis function models

and many more...

For example – we already have looked at basis function models \( y = w^T \Phi \)

The simple **linear** model has \( \Phi = [1, X]^T \): if we chose those \( X \) to be the recent past samples of \( y \), then this is just the **autoregressive model**

We can trivially extend this to a **non-linear basis**

\[
\Phi = [1, X, \phi_{\text{harmonics}}(X)]^T
\]

\[
\Phi = [1, X, \phi_{\text{Gaussians}}(X), \phi_{\text{harmonics}}(X)]^T
\]
Simple example

\[ y = \mathbf{w}^T \Phi \]

\[ \Phi = [1, \mathbf{X}, \phi_{\text{harmonics}}(\mathbf{X})]^T \]

Training data

Test (out of sample) data

forecasts
Method of analogues

A widely used method, especially in early **weather forecasting**
The following is the core assumption

If the **recent past** of a time series, is similar to historical sequences we have **previously seen** then the **future** will be similar to the 'futures' of those similar historic timeseries

\[
X = \sum_{i} w_{i} x_{i} \quad Y = \sum_{i} w_{i} y_{i}.
\]
Attractor distance

Mackey-Glass chaotic system
Method of embedding

(Takens) – we can reconstruct the attractor of a dynamical system using a tapped delay line – i.e. lagged versions of the time series

As we move on the attractor, we can use nearby trajectories to help forecast.
Improves performance

Using nearby trajectories

Using recent samples
Function Mappings - summary

These are **widely used** and **very flexible**. We run the risk of moving from *time series* to *machine learning* – and of course, there is a vast overlap.

The one problem of function mapping is that, without a lot of sophistication, the mapping we learn is **fixed**. Some ways around that:

1) rolling window – estimate a mapping using a rolling **subset of the data**
2) adaptive models – for example the **Kalman filter**

But now, let's go back though to the second prediction approach – that of **curve fitting**. Here we regress a function **through the time-varying values of the time series** and **extrapolate** (or **interpolate** if we want to fill in **missing values**) in order to **predict**
Curve fitting – is regression

We fit a function to the sequential data points, then extrapolate the function.
But... what form should the curve take?
...this is a problem with many possible solutions
Prior information may allow us to discriminate between solutions
The right model?

All these models explain the data equally well...
Maximum-likelihood solution

Severe underestimation of uncertainty away from data

\[
\frac{1}{\beta^*} = \mathbb{E}\{(y - t)^2\}
\]
Draws from posterior

Draws from function $p(t|x, w)p(w)$
The humble (but useful) Gaussian

\[ p(x_2|x_1 = \text{known}) \]

\[ p(x_1, x_2) \]

\[ p(x_2) \]

\[ p(x_1) \]
Observe $x_1$.
Extend to continuous variable
Probabilities over functions not samples

A “X” process is a distribution over a function space such that the pdf at any evaluation of the function are conditionally “X” distributed.

- Dirichlet Process [infinite state HMM]
- Indian Buffet Process [infinite binary strings] etc etc.
The Gaussian process model

- See the GP via the distribution

\[ p(y(x)) = \mathcal{N}(\mu(x), K(x,x)) \]

- If we observe a set \((x,y)\) and want to infer \(y^*\) at \(x^*\)

\[
p\left(\begin{bmatrix} y \\ y^* \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu(x) \\ \mu(x^*) \end{bmatrix}, \begin{bmatrix} K(x,x) & K(x,x^*) \\ K(x^*,x) & k(x^*,x^*) \end{bmatrix}\right)
\]

\[ p(y_*) = \mathcal{N}(m_*, C_*) \]

\[
m_* = \mu(x^*) + K(x^*,x)K(x,x)^{-1}(y - \mu(x)), \quad \sigma_*^2 = K(x^*,x^*) - K(x^*,x)K(x,x)^{-1}K(x,x^*)
\]
The beating heart...

What about these covariances though?

\[
K(x, x) = \begin{pmatrix}
  k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\
  k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\
  \vdots & \vdots & \ddots & \vdots \\
  k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n)
\end{pmatrix}
\]

Achieved using a kernel function, which describes the relationship between two points.

What form should this take though?
Kernel functions

The GP experience
Observe some data
Condition posterior functions on data
Simple regression modelling
Less simple regression
In a sequential setting
Simple comparison

GP

Spline
basis

 Harmonic
basis
The Kalman process revisited

In previous lectures we've seen how the Kalman updates produce an optimal filter under assumptions of linearity and Gaussian noise.

The Kalman process is one of an adaptive linear model, so if we regress from non-linear representation of the data then it becomes easy to develop a non-linear, adaptive model.

\[ \hat{y}[t] = \sum \limits_n w_n[t] \phi_n(Y_{past}) \]
Coping with missing values

Missing observations in the data stream $y[t]$

Can infer all or part of the missing observations vector as state-space model is linear Gaussian in the observations – simply replace the true observation with the inferred one.

If the model is for time-series prediction, then proxy observations are simply the most probable posterior predictions from the past time steps – this naturally leads to a sequential AR process.

$$\hat{y}[t] = \sum_{n} w_n[t] \hat{y}[t - n]$$

Could be directly observed or inferred
Brownian stream example

Component 1 missing

Component 2 missing
Weak correlation between streams

Markov correlation from blue to green

True value
Inferred value

Missing data regions
Comparison – synthetic Brownian data

GP

KF
**Application example**

Set of weather stations – local weather information
Comparison – air temperature

3 sensors
## Comparison: GP v KF

<table>
<thead>
<tr>
<th>State Space models</th>
<th>Gaussian Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computationally very efficient</td>
<td>Computationally demanding, but satisfactory for real-time systems</td>
</tr>
<tr>
<td>Infer posteriors over outcome variables</td>
<td>Infer posteriors over all variables, including hyper-parameters</td>
</tr>
<tr>
<td>Handle missing data and corruptions at all levels</td>
<td>Handling missing data and corruptions at all levels</td>
</tr>
<tr>
<td>Can extend to sequential / predictive decision processes with ease</td>
<td>More difficult to extend to decision processes at present</td>
</tr>
<tr>
<td>Active data requesting (request for observation or label)</td>
<td>Active data requesting</td>
</tr>
<tr>
<td>Prior knowledge of data <em>dynamics</em></td>
<td>Prior knowledge regarding nature of data <em>correlation length</em></td>
</tr>
</tbody>
</table>

Recent innovation sees intimate relationships between GPs and SSMs.
Particle filtering

In much of what we have looked at so far, we have assumed that the posterior distribution has some simple form – for example it is Gaussian.

All we then need to do is to infer the posterior mean and (co-)variance.

Why is this assumption useful?
- it means we can readily map the prior Gaussian to the posterior.

Many systems, though are not that simple – there may be multi-modalities and the posterior is non-Gaussian. Indeed it might even be that there is no simple parametric model that describes it (at least that we know about ahead of time).

Let’s think about a simple system that shows that this Gaussian assumption fails

\[ y[t] = y[t-1]^2 \]

If \( y[t-1] \) has a Gaussian posterior, used as prior to \( y[t] \), then we know that the prior cannot be conjugate with the posterior as \( y[t] \) cannot be Gaussian distributed.
So what's wrong with the EKF?

From Van der Merwe et al.
So we can propagate samples

So, rather than propagate the **sufficient statistics** (e.g. update the mean, variance) we can **sample from the posterior** over $y[t-1]$ and then transform each sample to obtain a **sample set which describes the distribution of $y[t]$**

**How do we sample?**

- Use **importance sampling**
- Leads to seeing **particle filters** as **Successive Important Resampling (SIR)** filters
Importance Resampling
An example

Consider our system state variable to evolve under a transform like

\[ a_{t+1} = F a_t + v_t \]  \hspace{1cm} \text{Diffusion process}

We can form the prior based on past observations

\[ p(a_t \mid X_{t-1}) = \int p(a_t \mid a_{t-1})p(a_{t-1} \mid X_{t-1}) \, da_{t-1} \]

We then observe the new datum \( x_t \)

\[ p(a_t \mid X_t) = Z^{-1} p(x_t \mid a_t) p(a_t \mid X_{t-1}) \]

1) Draw samples from \( p(a_{t-1} \mid X_{t-1}) \)
2) Propagate through \( a_{t+1} = F a_t + v_t \)
3) Get the importance weights \( q^n_t = \frac{p(x_t \mid a^n_{t-1})}{\sum_{k=1}^{N_p} p(x_t \mid a^n_{t-1})} \) and thence \( p(a_t \mid X_t) \)
4) iterate