Bayesian Methods for Data Modelling (Part 1)

Mike Tipping

January 24, 2008
EPSRC Winter School: Mathematics for Data Modelling
University of Sheffield
Outline (Part 1)

1. Prologue . . .
   - “Ockham’s Razor” and the “just right” model

2. Setting the Scene . . .
   - A simple linear regression problem
   - Least-squares approximation
   - Complexity control & regularisation

3. Bayesian Inference
   - Likelihood, priors & inference
   - MAP estimation
   - Marginalisation
   - “Ockham’s Razor” revisited
In the fourteenth century, William of Ockham proposed: “Pluralitas non est ponenda sine neccessitate” which literally translates as: “Entities should not be multiplied unnecessarily”

In a data modelling context, of all potential solutions to a given problem, we would ideally choose the simplest

Bayesian statistical inference automatically manages the trade-off between simplicity and solution accuracy
Consider that we have a binary (■/□) communication system

We have a fixed dictionary of symbols (strings of bit-1):
- ■■■■■■■■
- ■■■■■
- ■■■
- ■■
- ■

Messages are constructed by OR-ing an arbitrary number of bit-1 symbols in arbitrary positions within a field of bit-0’s

There may be transmission errors (independent inversion of bits)

We receive a binary sequence: \( t = ■■■■■■■■□□□□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
Some possible decodings:

<table>
<thead>
<tr>
<th>Decoding</th>
<th>Model $\mathcal{M}$</th>
<th>Error $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>■■■■■■■■■■■■■■■■■■■■■</td>
<td>6 $\times$ □</td>
</tr>
<tr>
<td>2</td>
<td>■■■■■■■■■■■■■■■■■■■■■</td>
<td>3 $\times$ □□ □□</td>
</tr>
<tr>
<td>3</td>
<td>■■■■■■■■■■■■■■■■■■■■■</td>
<td>□□□□□□ + □□□□□□□□□□</td>
</tr>
<tr>
<td>4</td>
<td>■■■■■■■■■■■■■■■■■■■■■</td>
<td>□□□□□□□□□□□□□□□□□□□</td>
</tr>
</tbody>
</table>

- Models 1–3 predict the sequence perfectly
- Model 4 is simplest, but requires introduction of bit errors
- Without any further assumptions, we can show that Decoding 3 is most probable ...
For each decoding, we calculate $p(t|M)$, the probability assigned to the sequence by the model with reference to all the other sequences that the model could potentially have decoded.

Giving:

| Decoding | Model $M$ | $\epsilon$ | # Sequences | $p(t|M)$ |
|----------|-----------|-------------|-------------|----------|
| 1        | 6 × □     | -           | 6! of $10^6$ | 0.0007   |
| 2        | 3 × □     | -           | 3! of $9^3$  | 0.0082   |
| 3        | □ □ □ □ + □ □ | - | 1 of $7 \times 9$ | 0.0159   |
| 4        | □ □ □ □ □ □ □ □ □ | 2 | 1 of $3 \times \binom{10}{2}$ | 0.0074   |

This simple example is “Bayesian inference in disguise”, and is exactly analogous to the way that Bayesian methods perform in more complex machine learning and data modelling tasks.
We have a set of ‘mystery’ data:

Truth: \( N = 15 \) samples synthesised from the function \( y = \sin(x) \) with added Gaussian noise of standard deviation 0.2

The ‘input’ variables are denoted \( x_n, n = 1 \ldots N \)

For each \( x_n \), there is an associated real-valued observation \( t_n \)
Model choice: parametric function $y(x; w)$

A linearly-weighted sum of $M$ fixed basis functions $\phi_m(x)$:

$$y(x; w) = \sum_{m=1}^{M} w_m \phi_m(x)$$

Example: Gaussian data-centred basis functions:

$$\phi_m(x) = \exp \left\{ -\frac{(x - x_m)^2}{r^2} \right\}$$

- a “radial basis function” (RBF) model
- $M = N = 15$ in this example
Goal: find \( \mathbf{w} \) such that \( y(x; \mathbf{w}) \) is a ‘good’ model

Start with a classic approach: *least-squares*, minimising:

\[
E_{LS}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left[ t_n - \sum_{m=1}^{M} w_m \phi_m(x_n) \right]^2
\]

If \( \Phi \) is the ‘design matrix’ such that \( \Phi_{nm} = \phi_m(x_n) \), and \( \mathbf{t} = (t_1, \ldots, t_N)^T \) then:

\[
\mathbf{w}_{LS} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}
\]
With $M = 15$ basis functions and only $N = 15$ examples, minimisation of squared-error leads to “over-fitting”:

How do we judge which model is “better”?

To estimate complexity, we must introduce some prior knowledge, preference, expectation, prejudice …
We typically prefer smoother functions, which typically have smaller weights \( \mathbf{w} \).

Augment the error function with a weight penalty term:

\[
E_{PLS}(\mathbf{w}) = E_{LS}(\mathbf{w}) + \lambda E_W(\mathbf{w})
\]

A conventional choice is the squared-weight penalty:

\[
E_W(\mathbf{w}) = \frac{1}{2} \sum_{m=1}^{M} w_m^2
\]

This conveniently gives the “penalised least-squares” estimate:

\[
\mathbf{w}_{PLS} = (\Phi^T \Phi + \lambda \mathbf{I})^{-1} \Phi^T \mathbf{t}
\]
The hyperparameter $\lambda$ controls the trade-off between quality of fit, $E_{LS}(\mathbf{w})$, and smoothness, $E_{W}(\mathbf{w})$. 

![Diagram showing training data and RBF models with different values of $\lambda$.](image)
Assess candidate values of $\lambda$ according to validation set data error.
Estimating $\lambda$ via Validation (2)

The diagram illustrates the normalised error against log $\lambda$ for training, validation, and test sets. The figure shows the error decreasing as $\lambda$ increases, with a sharp increase at a certain value of log $\lambda$. This indicates the optimal value of $\lambda$ for minimizing error.
Bayesian Inference: Basic Principles

- Define *prior* probability distributions over *all* model variables:
  - Inherently stochastic quantities (*e.g.* the observations $t$)
  - All *parameters* (*e.g.* $w, \sigma, \lambda$)
  - The model $\mathcal{M}$ itself (*e.g.* its type, structure, basis choice etc)

- Update these distributions in light of the data (Bayes’ rule!)

- *Integrate out* variables which are not directly of interest
  - Most required integrations are analytically intractable ✗

- Key features of the Bayesian approach:
  - A consistent way to deal with all sources of uncertainty ✓
  - An explicit framework for encoding prior knowledge ✓
  - Automatic implementation of “Ockham’s Razor” ✓
Data is a noisy sample from the underlying function:

\[ t_n = y(x_n; w) + \epsilon_n \]

Gaussian zero-mean noise model with variance \( \sigma^2 \):

\[ p(\epsilon_n | \sigma^2) = N(0, \sigma^2) \]

Assuming independence, the likelihood \( p(t|w, \sigma^2) \) of the data is:

\[
\prod_{n=1}^{N} p(t_n|w, \sigma^2) = \prod_{n=1}^{N} \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^{-1/2} \exp \left[ -\frac{\{t_n - y(x_n; w)\}^2}{2\sigma^2} \right]
\]

So “maximum-likelihood” \( \equiv \) “least-squares” here
Model complexity is controlled by specifying a prior distribution which expresses a “degree of belief” regarding appropriate values for \( w \) before observing the data.

A conventional choice is a zero-mean Gaussian:

\[
p(w | \alpha) = \left( \frac{\alpha}{2\pi} \right)^{M/2} \prod_{m=1}^{M} \exp \left\{ -\frac{\alpha}{2} w_m^2 \right\}
\]

This expresses a preference for smoother models by declaring smaller weights to be a priori more probable.

The strength of this preference is controlled by the shared inverse-variance hyperparameter \( \alpha \).
Bayesian Inference: Bayes’ Rule!

- Given the likelihood and the prior, we compute the posterior distribution over $w$ via Bayes’ rule:

$$p(w|t, \alpha, \sigma^2) = \frac{p(t|w, \sigma^2) p(w|\alpha)}{p(t|\alpha, \sigma^2)} = \frac{\text{likelihood} \times \text{prior}}{\text{normalising factor}}$$

- Here, the posterior is Gaussian: $p(w|t, \alpha, \sigma^2) = N(\mu, \Sigma)$ with

$$\mu = \left( \Phi^T \Phi + \sigma^2 \alpha I \right)^{-1} \Phi^T t$$

$$\Sigma = \sigma^2 \left( \Phi^T \Phi + \sigma^2 \alpha I \right)^{-1}$$
The “maximum a posteriori” (MAP) estimate for $w$ is the single most probable value under the posterior distribution $p(w|t, \alpha, \sigma^2)$.

For a Gaussian posterior, the maximum is equal to the mean:

$$w_{MAP} = \mu = (\Phi^T\Phi + \sigma^2\alpha I)^{-1}\Phi^Tt$$

Recall: $w_{PLS} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^Tt$

The MAP estimate is therefore identical to the penalised least-squares estimate re-parameterised with $\lambda = \sigma^2\alpha$. 
Over to Matlab ...
The MAP/PLS equivalence does *not* mean that the Bayesian framework is simply a re-interpretation of classical methods!

The key element of Bayesian inference is *marginalisation*, where we seek to integrate out all ‘nuisance’ variables, including $w$

This integration procedure automatically implements “Ockham’s Razor”: the intrinsic assignment of higher probability to “appropriately complex” models

We’ll exploit this to:
- robustly estimate the hyperparameter $\alpha/\lambda$ (next)
- selection of the model itself (later)
We should define priors over all variables, not just the weights $w$

Having defined priors $p(\alpha)$ and $p(\sigma^2)$, we apply Bayes’ rule:

$$p(w, \alpha, \sigma^2|t) = \frac{p(t|w, \sigma^2) p(w|\alpha) p(\alpha) p(\sigma^2)}{p(t)}$$

Not computable in closed form since the integral:

$$p(t) = \int p(t|w, \sigma^2) p(w|\alpha) p(\alpha) p(\sigma^2) \, dw \, d\alpha \, d\sigma^2$$

is not analytically tractable
We can’t compute $p(w, \alpha, \sigma^2 | t)$ analytically, so we desire a workable approximation.

We decompose the joint posterior as:

$$p(w, \alpha, \sigma^2 | t) \equiv p(w | t, \alpha, \sigma^2) \ p(\alpha, \sigma^2 | t)$$

The ‘weight posterior’ distribution $p(w | t, \alpha, \sigma^2)$ is tractable.

The ‘hyperparameter posterior’ $p(\alpha, \sigma^2 | t)$ must be approximated.
Type-II Maximum Likelihood

- Find the single ‘most probable’ values $\alpha_{\text{MP}}$ and $\sigma^2_{\text{MP}}$ under the posterior distribution:

$$p(\alpha, \sigma^2 | t) = \frac{p(t | \alpha, \sigma^2) p(\alpha) p(\sigma^2)}{p(t)}$$

- Assume log-uniform hyperpriors over $p(\alpha)$ and $p(\sigma^2)$

- Maximise $p(t | \alpha, \sigma^2)$, the marginal likelihood of the training data:

$$p(t | \alpha, \sigma^2) = \int p(t | w, \sigma^2) p(w | \alpha) \, dw$$

$$= (2\pi)^{-N/2} |\sigma^2 I + \alpha^{-1} \Phi \Phi^T|^{-1/2} \exp \left\{ -\frac{1}{2} t^T (\sigma^2 I + \alpha^{-1} \Phi \Phi^T)^{-1} t \right\}$$
Estimating $\alpha$ via Validation

$$\log \lambda$$

Normalised error

Training

Validation

Test

Mike Tipping

www.relevancevector.com

Jan 24, 2008 :: Sheffield
Estimating $\alpha$ via Marginal Likelihood
Ockham’s Razor revisited

- Marginalisation over $w$ implements “Ockham’s Razor” by rejecting models that are both too simple and too complex.

---

Diagram:
- **Marginal Model Probability** $p(t | \alpha)$
- **Space of all data sets** $t$ (increasing complexity)
- **High $\alpha$**
- **Medium $\alpha$**
- **Low $\alpha$**

*Noisy Sine Data*
End of Part One . . .

Don’t go away, we’ll be right back
Rules of Probability

- Product rule:
  \[ p(a, b) = p(a|b) p(b) = p(b|a) p(a) \]

- More generally:
  \[ p(a, b|c) = p(a|b,c) p(b|c) = p(b|a,c) p(a|c) \]

- Rearranging gives Bayes’ rule:
  \[ p(a|b,c) = \frac{p(b|a,c) p(a|c)}{p(b|c)} \]

- Sum (integral) rule:
  \[ p(b|c) = \int_{-\infty}^{\infty} p(a, b|c) \, da \]