Searching for 'optimal' inputs with an empirical regression model

David J.C. MacKay
University of Cambridge
Cavendish Laboratory, Madingley Road, Cambridge CB3 0HE
mackay@mrao.cam.ac.uk

November 25, 1997 — Version 1.2

Abstract

These notes review the idea of non–linear data modelling then explain the rationale behind the objective functions that can be used in the Cambridge Gaussian Process software package when optimizing the input variables.

1 Empirical regression models

Let us assume that we have gathered a data set $\{\mathbf{x}, t\}$, where $\mathbf{x}$ are the input variables, assumed to be measured accurately, and t is the measured 'target' variable, which is assumed to be a noisy version of an underlying output variable $y(\mathbf{x})$.

We represent the inference of this unknown function $y(\mathbf{x})$ from the data using Bayes theorem:

$$P\left(y(\mathbf{x})|\{t\}\right) = \frac{P(\{t\}|y(\mathbf{x}))P(y(\mathbf{x}))}{P(\{t\})}. \tag{1}$$

The left hand side, $P\left(y(\mathbf{x})|\{t\}\right)$, is the **posterior probability** of the function $y(\mathbf{x})$ given the data. On the right hand side, the first factor is the **likelihood function** $P(\{t\}|y(\mathbf{x}))$. This specifies the probability of obtaining the observed data set $\{t\}$ if the true underlying function were $y(\mathbf{x})$, that is, it specifies the noise model. The simplest noise model assumes that the noise is additive and independent from sample to sample, and Gaussian distributed. The use of least-squares estimation methods corresponds directly to an assumption of independent Gaussian distributed noise. The second factor is the **prior** $P(y(\mathbf{x}))$. This expresses our prior assumptions about the nature of the function $y(\mathbf{x})$, for example, an assumption that $y(\mathbf{x})$ is a continuous and smooth function of $\mathbf{x}$. Clearly, without some such assumptions it is impossible to do regression. The key do doing good empirical modelling is to use a carefully chosen likelihood function and a carefully chosen prior which capture prior beliefs and uncertainties about the noise process and the underlying function $y(\mathbf{x})$. The normalizing constant $P(\{t\})$ is called the **evidence** for the model. It does not depend on $y(\mathbf{x})$ so is not important if our hypothesis space is fixed.

But if there are aspects of our prior and likelihood function that are uncertain, then the evidence can act as a guide towards the most probable priors and the most probable likelihood function.

We may be interested in the following issues.

1. How to **represent** the function $y(\mathbf{x})$. In order to obtain predictions that we can trust, our representation should be capable of describing all complex non-linear functions that we believe might be the case.

2. How to describe a **prior** distribution over $y(\mathbf{x})$ that captures the concept of uncertain **relevance** of input variables.

3. How to **implement** the computations corresponding to the inference of equation (1), and how to represent the inferences (*e.g.*, predictive mean and error bars).

4. Having inferred $y(\mathbf{x})$ with error bars, we might also be interested in two types of **decision** problem.

   (a) The **inversion** problem — selecting input values so as to achieve a desired output value.

   (b) The **experimental design** or **active learning** problem — selecting input values for new experiments in order to learn useful information about $y(\mathbf{x})$.

   Both of these decision problems require that we (implicitly or explicitly) define **utility functions** that measure, for example, how costly inaccuracy is, and how informative a measurement is.

2 Representation and Priors

Let us discuss the question of representation and priors together. There are two approaches to representing a function $y(\mathbf{x})$. One is the **parametric** aproach where y is written as an explicit function of $\mathbf{x}$, $y(\mathbf{x};\mathbf{w})$, parameterized by parameters $\mathbf{w}$. Neural networks are an example of a parametric model, as are radial basis functions. A standard method for putting a prior on y is to put a prior on the parameters $\mathbf{w}$, for example, a Gaussian prior. (For reading on Gaussian priors applied to neural networks see (MacKay 1991; MacKay 1992a; MacKay 1992d; MacKay 1992b; MacKay 1996; MacKay 1995; Neal 1996).)

In a **nonparametric** approach there are no parameters describing the function $y(\mathbf{x})$, but there is some procedure for predicting $y(\mathbf{x}_{N+1})$ given the training data. Examples of non-parametric models are cubic splines interpolation, and **Gaussian processes**. Splines correspond to a model in which the prior probability distribution over the function $y(x)$ (Kimeldorf and Wahba 1970) is:[1]

$$\log \mathcal{P}(y(x)|\alpha, \mathcal{H}_1) = -\frac{1}{2}\,\alpha \int dx\, [y^{(p)}(x)]^2 + \text{const}, \tag{2}$$

where $y^{(p)}$ denotes the pth derivative of y, and $p = 2$ for cubic splines.

[1] Strictly this prior is improper since addition of an arbitrary polynomial of degree $p-1$ to $y(x)$ is not constrained. It can be made proper by adding terms corresponding to boundary conditions to (2).

2.1 Gaussian processes

The Gaussian process model is now our preferred model for non-linear regression problems. In a Gaussian process (Williams 1995; Williams and Rasmussen 1996),[2] we model the joint distribution of $\{y(\mathbf{x}^{(n)})\}_{n=1}^{N}$ with a Gaussian distribution

$$P\left(\{y(\mathbf{x}^{(n)})\}_{n=1}^{N}\right) = \frac{1}{Z}\exp\left(-\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y}\right), \tag{3}$$

where $\mathbf{C}$ is an appropriate positive definite covariance matrix which is a function of the input coordinates $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$, and possibly of some hyperparameters. Most parametric models and non-parametric regression models are in fact special cases of Gaussian processes, with the covariance matrix depending on the details of the model. Efficient methods for implementing Gaussian processes are described in Gibbs and MacKay (1996). Gaussian processes have the advantage that predictive distributions given data are obtained by simple matrix operations. (These matrix operations are the equivalent of two operations in the case of a standard implementation of a parametric model — the parameter adaptation process and the computation of error bars on those parameters.)

2.2 Hyperparameters

Lurking in most prior probability distributions are 'hyperparameters', such as α in (2), which have the role of controlling the 'complexity' of functions drawn from the prior, and thus the complexity of the inferred function. The values of such hyperparameters are important in controlling the nature of the predictions given by the model. Hyperparameters can be controlled using Bayesian methods. In many problems it is advantageous to have multiple hyperparameters that control different aspects of complexity. For example, if we are unsure how 'relevant' each input is in a non-linear regression, we can have for each input variable one hyperparameter that quantifies the lengthscale on which the output varies significantly as a function of that input variable. These unknown lengthscales can then be inferred from the data. This 'automatic relevance determination' method is available when we use either neural networks or Gaussian processes.

3 Implementation method

There are deterministic and Monte Carlo implementation methods (Neal 1993) for Bayesian models. Both have advantages.

4 Decision problems

In order to make decisions, we need to have not only a probabilistic model but also a **utility function**. If there are possible actions a and possible states of the world s with probability distribution, given the current information, $P(s)$, then we need a utility

[2] Gaussian processes have also been used in geophysical data modelling.

function $U(a, s)$ which says what the utility of action a is if the state of the world is in fact s. The optimal action given P and U is then

$$a_{\text{opt}} = \text{argmax}\, \bar{U}(a) \tag{4}$$

where

$$\bar{U}(a) = \int ds\, P(s) U(a, s). \tag{5}$$

Decision theory is thus 'trivial' once a utility function and probability distribution are given.

What should the utility function be?

4.1 Utility functions for experimental design

I have studied the problem of active learning in neural networks a few years ago and derived a few experimental design objective functions (MacKay 1992c). I learnt two things from this work:

1. The choice of objective function must be made carefully to avoid getting silly results. If one wants an objective function that sensibly measures how informative a new measurement would be, it is necessary to define a **region of interest** about which information is desired.

2. The objective functions I derived seem typically to be multimodal so searching for the most informative measurement may be a non-trivial task.

4.2 Utility functions for inversion and selection of 'optimal' inputs

Imagine we want to select an input that produces a desired output; or maybe we want to select an input that is likely to produce the **maximum** output. What utility functions describe these situations?

In the notation introduced above, the choice of action a is a choice of input $\mathbf{x}$; the state of the world s is the true function y. The utility function is $U(\mathbf{x}, y)$ which, we will assume, has the form $u(y(\mathbf{x}))$, so that

$$\bar{U}(\mathbf{x}) = \int dy_{\mathbf{x}}\, P(y_{\mathbf{x}}) u(y_{\mathbf{x}}). \tag{6}$$

In general there might be an additional cost associated with the choice of $\mathbf{x}$ — perhaps $\mathbf{x}$ represents the choice of ingredients in some process and y represents the quality of the product; the economic utility takes into account both the cost of the ingredients $\mathbf{x}$ and the expected quality of the product. But if we assume that these two terms are additive then there is no need to consider them together.

4.3 Utility function for inversion

We want the output y to be as close as possible to z. A utility function that does the trick is

$$u_1(y) = (y - z)^2. \tag{7}$$

If we write $P(y_{\mathbf{x}})$ as a Gaussian

$$P(y_{\mathbf{x}}) = \text{Normal}(y_{\mathbf{x}}; \mu_{\mathbf{x}}, \sigma^2_{\mathbf{x}}) \tag{8}$$

[where $\sigma^2_{\mathbf{x}}$ describes the uncertainty in the value of $y(\mathbf{x})$] then the expected utility is

$$\bar{U}_1(\mathbf{x}) = (\mu_{\mathbf{x}} - z)^2 + \sigma^2_{\mathbf{x}}. \tag{9}$$

It is easy in the case of Gaussian processes to evaluate the derivative of this objective function with respect to $\mathbf{x}$.

4.4 Utility function for maximization

We want the output y to be as large as possible. The most obvious utility function, but one which I think is not appropriate, is:

$$u_2(y) = y. \tag{10}$$

The expected utility is

$$\bar{U}_2(\mathbf{x}) = \mu_{\mathbf{x}}, \tag{11}$$

which does not take into account the error bars. Is this appropriate? The user must decide, but I would suggest that if there is some value of $\mathbf{x}$ where the error bars on y are enormous, and the mean happens to be a little higher than elsewhere, it seems unlikely that this is the optimal location. An objective function that takes into account error bars seems desirable.

4.4.1 A concave utility function

If y is not known to be positive, it may be that an appropriate utility function is

$$u_3(y) = -\exp(-\beta y) \tag{12}$$

where β is a dimensional constant that the user must specify. This utility function is concave downwards and so (by Jensen's inequality) increasing variance in y leads to a decrease in expected utility.

The expected utility, given a Gaussian distribution for y, is

$$\bar{U}_3(\mathbf{x}) = \int dy \, \text{Normal}(y_{\mathbf{x}}; \mu_{\mathbf{x}}, \sigma^2_{\mathbf{x}})[-\exp(-\beta y)] = -\exp\left[-\beta\mu_{\mathbf{x}} + \frac{1}{2}\beta^2\sigma^2_{\mathbf{x}}\right] \tag{13}$$

So from the point of view of optimization, the objective function might as well be:

$$\bar{V}_3(\mathbf{x}) \equiv \frac{1}{\beta}\left\{-\log[-\bar{U}_3(\mathbf{x})]\right\} = \mu_{\mathbf{x}} - \frac{1}{2}\beta\sigma^2_{\mathbf{x}}. \tag{14}$$

This makes complete sense. We want big μ, but we also want small σ^2. The user chooses the trade-off between these by setting β.

It is easy in the case of Gaussian processes to evaluate the derivative of this objective function with respect to $\mathbf{x}$.

4.4.2 Special case: y positive

If y is in fact known to be positive, then it may be that an appropriate utility function is

$$u_4(y) = \log(y/\gamma) \tag{15}$$

where γ is a dimensional constant that the user need not specify [sic]. This utility function is concave downwards and so (by Jensen's inequality) increasing variance in y leads to a decrease in expected utility.

This utility function says, for example, that a chance of a 100% increase in y is about equal and opposite in value to a chance of a 50% decrease in y.

We will derive the expected utility below.

4.4.3 General case: other concave objective functions

Imagine that we use a concave utility function with a Taylor expansion, around some convenient value of μ,

$$u(y) = u(\mu) + \alpha(y - \mu) - \beta(y - \mu)^2 \ldots \tag{16}$$

Then the expected utility is (assuming we can neglect subsequent terms)

$$\bar{U}(\mathbf{x}) \simeq u(\mu_{\mathbf{x}}) - \beta\sigma_{\mathbf{x}}^2. \tag{17}$$

For example, in the case of $u_4 = \log(y/\gamma)$, $\beta = \frac{1}{2}\frac{1}{\mu^2}$, so

$$\bar{U}_4(\mathbf{x}) \simeq \log(\mu_{\mathbf{x}}/\gamma) - \frac{1}{2}\frac{\sigma_{\mathbf{x}}^2}{\mu^2}. \tag{18}$$

4.4.4 Other functions of the input

Since it is easy to visualize, we have also included the objective function

$$U(\mathbf{x}) \simeq \alpha\mu_{\mathbf{x}} + \beta\sigma_{\mathbf{x}}, \tag{19}$$

which, if α and β are both 1 defines the upper one–sigma error bar on the predictions. So minimizing this U finds the input for which the upper error bar is lowest.

4.5 Discussion

Whether these objective functions can be optimized using gradient descent will clearly depend on the problem at hand. We have implemented all the above objective functions in our Gaussian process software.

Appendix

Jensen's inequality. If f is a convex function (convex upwards) and x is a random variable then:

$$E\left[f(x)\right] \geq f\left(E[x]\right), \tag{20}$$

where E denotes expectation. If f is strictly convex and $E\left[f(x)\right] = f\left(E[x]\right)$, then x is a constant (with probability 1).

References

Gibbs, M. N., and MacKay, D. J. C., (1996) Efficient implementation of Gaussian processes for interpolation. in preparation.

Kimeldorf, G. S., and Wahba, G. (1970) A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. *Annals of Mathematical Statistics* **41** (2): 495–502.

MacKay, D. J. C., (1991) *Bayesian Methods for Adaptive Models*. California Institute of Technology dissertation.

MacKay, D. J. C. (1992a) Bayesian interpolation. *Neural Computation* **4** (3): 415–447.

MacKay, D. J. C. (1992b) The evidence framework applied to classification networks. *Neural Computation* **4** (5): 698–714.

MacKay, D. J. C. (1992c) Information based objective functions for active data selection. *Neural Computation* **4** (4): 589–603.

MacKay, D. J. C. (1992d) A practical Bayesian framework for backpropagation networks. *Neural Computation* **4** (3): 448–472.

MacKay, D. J. C. (1995) Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems* **6**: 469–505.

MacKay, D. J. C. (1996) Bayesian non-linear modelling for the 1993 energy prediction competition. In *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, ed. by G. Heidbreder, pp. 221–234, Dordrecht. Kluwer.

Neal, R. M. (1993) Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG–TR–93–1, Dept. of Computer Science, University of Toronto.

Neal, R. M. (1996) *Bayesian Learning for Neural Networks*. Number 118 in Lecture Notes in Statistics. New York: Springer.

Williams, C. K. I., (1995) Regression with Gaussian processes. To appear in Annals of Mathematics and Artificial Intelligence.

Williams, C. K. I., and Rasmussen, C. E. (1996) Gaussian processes for regression. In *Advances in Neural Information Processing Systems 8*, ed. by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo. MIT Press.