SEQUENTIAL BAYESIAN COMPUTATION OF LOGISTIC REGRESSION MODELS

Mahesan Niranjan

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, England
niranjan@eng.cam.ac.uk

ABSTRACT

The Extended Kalman Filter (EKF) algorithm for identification of a state space model is shown to be a sensible tool in estimating a Logistic Regression Model sequentially. A Gaussian probability density over the parameters of the Logistic model is propagated on a sample by sample basis. Two other approaches, the Laplace Approximation and the Variational Approximation are compared with the state space formulation. Features of the latter approach, such as the possibility of inferring noise levels by maximising the 'innovation probability' are indicated. Experimental illustrations of these ideas on a synthetic problem and two real world problems are discussed.

1. INTRODUCTION

The logistic regression model is widely used in Bayesian Inference problems. There are several examples of experimental illustrations of new, and supposedly more powerful, ideas that perform worse than the simple logistic model. In Bayesian graphical modelling, the logistic is often used as the parametric model of conditional probabilities.

Many problems in inference are characterised by data that arrives sequentially. This is particularly true in applications in time series analysis and control of dynamical systems. Sequential estimation might be of use in classification problems too if one is in a nonstationary environment and is interested in online reestimation, application and refinement of the models. Computational simplicity in the form of not having to store all the data might also be an additional motivating factor for sequential learning.

In this paper, I consider the logistic regression model, with Bayesian training performed sequentially. I show how Bayesian computations can be performed sequentially in an Extended Kalman Filtering setting. Taking the logistic function through the EKF update equations results in simple update algorithms for the parameter mean and covariance matrix. These updates turn out to be almost identical in form, and even closer in practice, to two other approaches to the same problem reported in recent machine learning literature. These are from Spiegelhalter & Lauritzen (1990) and Jaakkola and Jordan (1996) [13, 5]. The main contribution of this paper is to points out the simplifications used in the EKF framework to achieve update equations so similar to the above two. approaches. We immediately see, and I show this on experimental simulations, that one might be able to do better by relaxing the simplifications forced in the EKF.

2. SEQUENTIAL LOGISTIC REGRESSION

Following the notation used in Jaakkola *et al.*, the logistic regressor is given by

$$p\left(s\ \boldsymbol{\theta}\right)\ =\ g\left(\left(2s-1\right)\boldsymbol{\theta}^{\mathrm{t}}\boldsymbol{x}\right),$$

where $g(.)$ is the sigmoid $g(\alpha)\ =\ 1/(1\ +\ \exp(-\alpha)$. The uncertainty in parameters $\boldsymbol{\theta}$ is represented as a Gaussian probability density function $\mathcal{N}\left(\boldsymbol{\mu},\ \boldsymbol{\Sigma}\right)$. In sequential estimation, we consider examples $\{\boldsymbol{x}_n,\ s_n\}$ arriving one at a time, as n takes values $1, 2, \ldots$. The task at the arrival of each example is to compute a Gaussian *posterior* probability distribution. We denote the parameters in this *posterior* distribution by $\mathcal{N}\left(\boldsymbol{\mu}_p,\ \boldsymbol{\Sigma}_p\right)$. Computing the true *posterior* is analytically not feasible. We assume the Gaussian approximation to the *posterior* is sufficient for our purposes.

2.1. Laplace Approximation

Spiegelhalter and Lauritzen derive an update algorithm for the case where the prior is Gaussian, *i.e.* $\mathcal{N}\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\right)$, by making the Laplace approximation of fitting a quadratic function to the Log *posterior.*

The corresponding update equations are

$$\begin{aligned}\boldsymbol{\Sigma}_p^{-1} &= \boldsymbol{\Sigma}^{-1}\ +\ g(1-g)\ \boldsymbol{x}_n\ \boldsymbol{x}_n^{\mathrm{t}}\\ \boldsymbol{\mu}_p &= \boldsymbol{\mu}\ +\ (s_n-g)\boldsymbol{\Sigma}_p\ \boldsymbol{x}_n\end{aligned}$$

2.2. Variational Apprximation

The variational approximation idea, Jaakkola *et al.* (1996), is to define a convex function that produces a bound on the likelihood. This bounding function is defined with a tuning parameter in it. In an iterative manner, performed in an EM framework, one alternates between maximisation of the bounding function and setting of the tuning parameters. This leads to the following update equations

$$\begin{aligned}\boldsymbol{\Sigma}_p^{-1} &= \boldsymbol{\Sigma}^{-1}\ +\ 2\lambda(\xi)\ \boldsymbol{x}_n\ \boldsymbol{x}_n^{t}\\ \boldsymbol{\mu}_p &= \boldsymbol{\Sigma}_p\left[\boldsymbol{\Sigma}^{-1}\ \boldsymbol{\mu}\ +\ (s_n-1/2)\ \boldsymbol{x}_n\right]\end{aligned}$$

The tuning parameter in this variational approximation is defined in a particular functional form

$$\lambda(\xi)\ =\ [0.5-g(\xi)]/2\xi$$

It can be optimised at the arrival of every example by running the EM algorithm. The update for ξ is given by

$$\xi^2\ =\ \boldsymbol{x}^{\mathrm{t}}\boldsymbol{\Sigma}_p\boldsymbol{x}\ +\ \left(\boldsymbol{x}^{\mathrm{t}}\boldsymbol{\mu}_p\right)^2$$

The update equations in these two formulations are very similar, in particular, it turns out that the weighting for the inverse covariance updates, $g(1-g)$ and $2\lambda\,(\xi)$ only take values between 0 and 0.25.

2.3. Extended Kalman Filter

Consider the simplified dynamical system given by the following equations:

$$\begin{aligned} \boldsymbol{\theta}(n+1) &= \boldsymbol{\theta}(n) + \boldsymbol{w}(n) \\ y(n) &= f\left(\boldsymbol{\theta}(n),\, \boldsymbol{x}(n)\right) + v(n) \end{aligned}$$

For simplicity, the evolution of the parameter vector $\boldsymbol{\theta}$, is taken as a random walk model, and we consider a single output case. Mainstream literature on Kalman filtering is more general than this, allowing for some known dynamics of the state vector, deterministic exogenous inputs etc. We stay with the simple model for brevity, extending any ideas reported here to the more general case is straightforward. The process noise $\boldsymbol{w}(n)$ has covariance matrix $\boldsymbol{Q}$, and the measurement noise has variance R, both assumed Gaussian. In general these parameters hold the key to the success, or failure, of applying the Kalman filtering framework to any practical problem. In some applications, control for example, one might have knowledge about these noise processes via additional measurements. In parameter estimation problems, such as the one considered here, there are systematic ways of tuning these noise parameters [6, 3]. We return to this point later.

The update equations for linear output case, *i.e.* $y(n) = \boldsymbol{\theta}^{\mathrm{t}}\,\boldsymbol{x} + v(n)$, consist of the elegant structure of a sequence of prediction and correction phases given by the following equations.

Prediction	$\boldsymbol{\theta}(n\|n-1) = \boldsymbol{\theta}(n-1\|n-1)$ $\boldsymbol{P}(n\|n-1) = \boldsymbol{P}(n-1\|n-1) + \boldsymbol{Q}$
Data at Time n	$\{\boldsymbol{x}(n),\, y(n)\}$
Error / Innovation	$e(n) = y(n) - \boldsymbol{\theta}^{\mathrm{t}}(n\|n-1)\,\boldsymbol{x}(n)$
Kalman Gain	$\boldsymbol{k}(n) = \frac{\boldsymbol{P}(n\|n-1)\,\boldsymbol{x}(n)}{\left\{R + \boldsymbol{x}^{\mathrm{t}}(n)\boldsymbol{P}(n\|n-1)\boldsymbol{x}(n)\right\}}$
Correction	$\boldsymbol{\theta}(n\|n) = \boldsymbol{\theta}(n\|n-1) + \boldsymbol{k}(n)e(n)$ $\boldsymbol{P}(n\|n) = \boldsymbol{P}(n\|n-1) - \boldsymbol{k}(n)\boldsymbol{x}^{\mathrm{t}}(n)\boldsymbol{P}(n\|n-1)$

In the case of a nonlinear model, such as a neural network, we expand the function $f\,(.)$ by Taylor series in the space of the parameters $\boldsymbol{\theta}$, and truncate it. It is common to use only terms upto the first order, In the options price tracking application [9], I found extending to the second order is well worth the effort. Many authors have looked at training a Neural Network in this setting. In doing this, the dynamical system remains unchanged, and the function $f\,(.)$ becomes the neural network. The update equations will have $\boldsymbol{x}$ replaced by the gradient vector of the output $f_{\theta}y(n)$ with respect to the unknown parameters. This, of course, can be calculated by error propagation. In work reported in Kadirkamanathan & Niranjan [7], we considered a Radial Basis Functions setting. Puskorius & Feldkamp [11] reports a highly successful application that uses a large Recurrent Neural Network.

While the material presented so far is all straightforward and at least a couple of decades old, what makes this framework interesting today is the flexibility it offers to a number of problems addressed by the Neural Network community, in the Signal Processing context in in particular. These have to do with regularisation in a sequential (or on-line) setting. When we restrict ourselves to looking at one example at a time, many of the well known tricks in training neural networks, such as cross validation to select learning parameters and model size, or bootstrap to deal with model uncertainty are no longer available to us.

That a Kalman filter can become handy in this environment can be seen from a Bayesian derivation of the filter. Recall that we are interested in propagating a Gaussian density in the parameter space. At time n, we can write Bayes rule as follows:

$$p\left(\boldsymbol{\theta}(n)|Y(n)\right) = \frac{p\left(y(n)|\boldsymbol{\theta}(n)\right)\; p\left(\boldsymbol{\theta}(n)|Y(n-1)\right)}{p\left(y(n)|Y(n-1)\right)}$$

Here, $p\left(\boldsymbol{\theta}(n)|Y(n)\right)$ is the *posterior* probability distribution, having seen all the data for $n = 1, ..., N$.

The denominator term in the above expression is known as the *innovation probability*. A Bayesian way of describing this quantity is that this represents the only additional piece of information available in the new data item, having absorbed all knowledge upto that point in time via the dynamical system as well as integrating out all uncertainties of the parameters. From a practical perspective, this term gives us a handle to systematically tune the unknown noise processes. Jazwinski has derived a number of algorithms to estimate these noise parameters by maximising the innovation probability [6]. These are also applicable to the regularisation of neural networks in a sequential training framework [3] The above report also describes the similarities to recent Bayesian ideas, or the so called 'evidence procedure', described in [1], Ch. 10.

2.4. Logistic through the EKF

Let us take the logistic regression through the extended Kalman filter. For the gradient of the logistic output $y(n)$ with respect to the parameter vector $\boldsymbol{\theta}$, we have

$$\boldsymbol{f}_{\theta} = g(1-g)\boldsymbol{x}_n$$

Substituting in the the covariance update equation, we have

$$\begin{aligned} P(n|n) &= P(n|n-1) - \frac{P(n|n-1)\boldsymbol{f}_{\theta}\boldsymbol{f}_{\theta}^{\mathrm{t}}P(n|n-1)}{R + \boldsymbol{f}_{\theta}^{\mathrm{t}}P(n|n-1)\boldsymbol{f}_{\theta}} \\ &= P(n|n-1) - \frac{P(n|n-1)\boldsymbol{x}\boldsymbol{x}^{\mathrm{t}}P(n|n-1)}{R/\left(g(1-g)\right)^2 + \boldsymbol{x}^{\mathrm{t}}P(n|n-1)\boldsymbol{x}} \end{aligned}$$

Setting $R = g(1-g)$ and applying matrix inversion lemma, we have

$$P^{-1}(n|n) = P^{-1}(n|n-1) + g(1-g)\boldsymbol{x}\boldsymbol{x}^{\mathrm{t}}$$

Taking this one step further, we can look at what is known as the iterated Extended Kalman filter, which allows one to make repeated local linearisations at the operating point (Bar Shalom & Fortman, 1988). For this we need the following iterations:

$$\begin{aligned} \boldsymbol{\theta}^{\mathrm{i}+1}(n|n) &= \boldsymbol{\theta}^{\mathrm{i}}(n|n) + \frac{1}{R}P^{\mathrm{i}}f_{\theta\theta}^{\mathrm{i}\,\mathrm{t}}(n)\left\{y(n) - f\left(\boldsymbol{\theta}^{\mathrm{i}}(n|n)\right)\right\} \\ &\quad - P^{\mathrm{i}}(n|n)P^{-1}(n|n-1) \\ &\quad \left\{\boldsymbol{\theta}^{\mathrm{i}}(n|n) - \boldsymbol{\theta}^{\mathrm{i}}(n|n-1)\right\} \\ P^{\mathrm{i}}(n|n) &= P(n|n-1) \\ &\quad - \left[R + f_{\theta}^{\mathrm{i}}(n)P(n|n-1)f_{\theta}^{\mathrm{i}\,\mathrm{t}}(n)\right]^{-1} \\ &\quad f_{\theta}^{\mathrm{i}}(n)P(n|n-1) \end{aligned}$$

where $\boldsymbol{f}_\theta$ is the gradient and $f_{\theta\theta}$ is the Hessian of the output with respect to the state vector. The superscript i refers to the iteration.

Taking the logistic model through these equations, and looking for the fixed points of the state update equation above gives the following results:

$$\begin{aligned}[P(n|n)]^{-1} &= [P(n|n-1)]^{-1} + g^i(1-g^i)\boldsymbol{x}\boldsymbol{x}^{\mathrm{t}} \\ \theta(n|n) &= \theta(n|n-1) + (y(n) - g^{\mathrm{i}})P(n|n-1)\boldsymbol{x}.\end{aligned}$$

These equations are identical to the update equations derived by the Laplace approximation. We have arrived at the same update equations in a roundabout way. What is interesting about this result is the set of simplifications we had to do along the way. Recall that we used

- local linearisation of the output equation at the operating point, and
- set the measurement noise variance to $g(1-g)$ at every example.

Obviously, then, there can be situations where we could do better with a Kalman filtering framework by not making these restrictions. For instance we could do a quadratic approximation about the operating point, or set the noise variance to some other value to give us greater flexibility.

3. ILLUSTRATIONS

3.1. Synthetic Problem

The first test problem considered here is a simple classification task in two dimensions, where the two classes were Gaussian distributed with distinct means and equal covariance matrices. For this case, we know that the Bayes optimal class boundary is a straight line and the corresponding posterior probability is a logistic function. When estimating a logistic model to classify data from such a problem, we have a perfect match between the optimum solution and the functional form of the model. As one might expect in such a case, all three algorithms converge to the correct answer after presentation of a small number of examples. Fig. 3.1 shows the typical behaviour; after the presentation of about 25 examples, the solutions are almost identical.

3.2. Australian Credit Data

This dataset originates from Quinlan [12, 10] and can be obtained from the STATLOG project archives [1]. The problem comes from a credit card applications domain and the features are a mixture of continuous and discrete valued variables.

Fig. 3.2 shows the effect of automatically setting the measurement noise variance, R, to maximize the innovation probability $p\left(y(n)|Y(n-1)\right)$. Formulae for these calculations are given in De Freitas *et al.* [3]. The test set classification error is plotted as a function of iteration. We see that, though asymptotically both the Laplace approximation and the extended Kalman algorithm produce results similar to the batch solution, the Kalman algorithm achieves a much smoother convergence. Examples in the training sequence that are large outliers from what has been seen upto some point in time, tend to cause sudden jumps in the solution produced by the Laplace approximation.

[1] `http://blanche.polytechnique.fr/www.eark/ftp_site/ML_Repository/statlog`

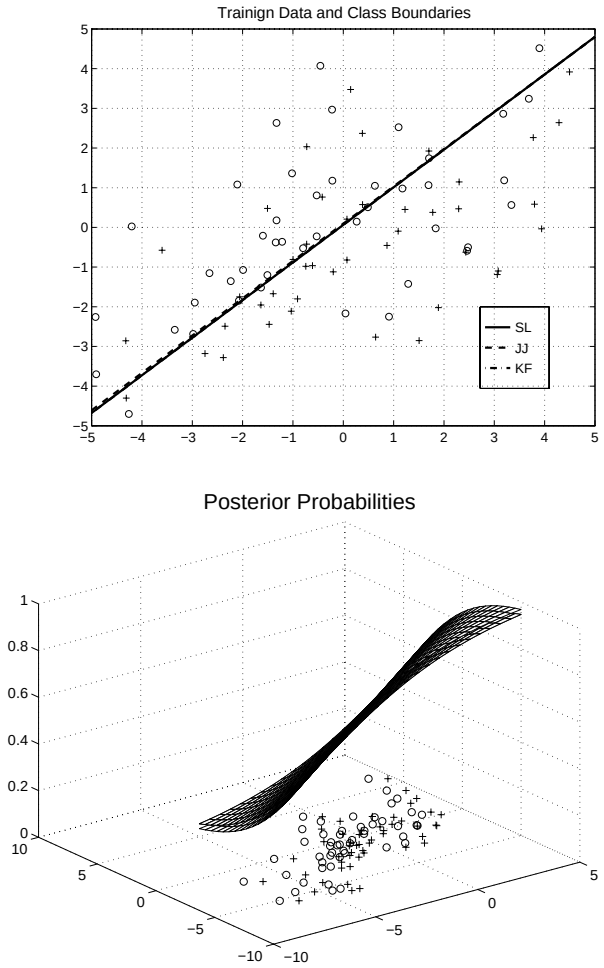

Figure 1: Class boundaries and posterior probabilities for a simple two dimensional example

3.3. Medical Risk Prediction

Lovell *et al.* (1997) describe the *QAMC* project, which is an attempt to predict the risk of adverse outcome in pregnancy using a large number of features forming the patient profile. This is a very large database of 750000 different patients. The features constitute a 44 dimensional binary space (some discrete values were quatised to give a binary representation), with a sparse occupancy; *i.e.* only a small subset of the patient profile bins are represented. Using the area under the Receiver Operating Characteristic (ROC) curve as an objective measure, Lovell *et al.* report sequential forward selection procedure for variable selection [8]. The interesting result from the study is that the Expected Attainable Discrimination, defined for this problem, can be very closely approximated by the Logistic Model. Similar good performance from the logistic on large medical datasets have been reported by others too.

A comparison of sequentially training a lotistic with the Variational Approximation scheme and the EKF formulation with automatically setting the measurement noise variance R is shown in Fig. 3. The upper graph is from the EKF, achieving a small but

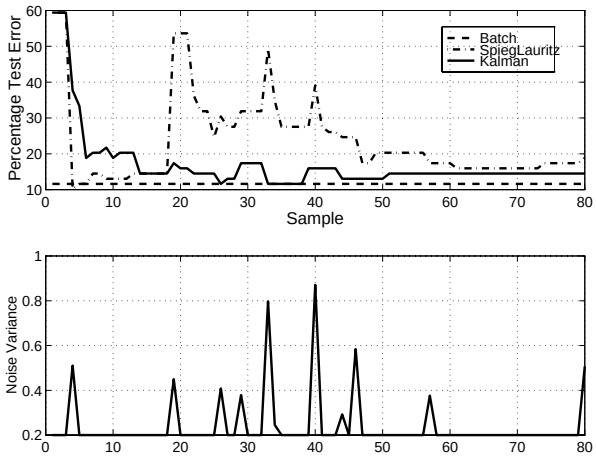

Figure 2: Comparing the behaviour of the Laplace approximation and the extended Kalman filter with measurement noise variance R inferred from data. On the Australian credit card problem, the upper plot shows the classification performance on the test data at the presentation of each example. The lower plot shows the estimated value of R. Note that in a number of places where R is increased by the Kalman algorithm are cases that cause the Laplace approximation to make sudden jumps. Increased R corresponds to not 'trusting' the the new data much, hence smaller update.

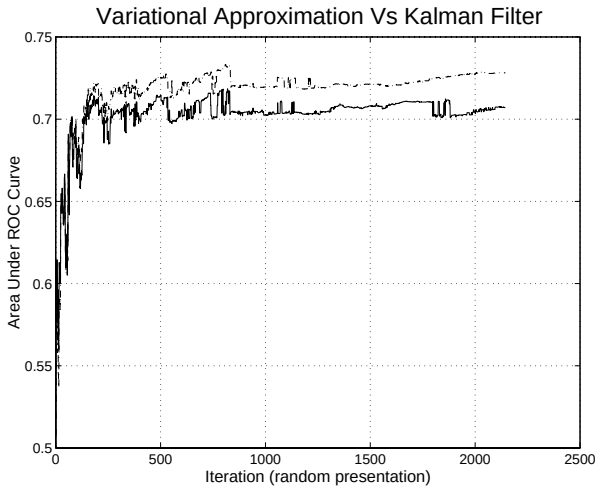

Figure 3: A comparison of the variational approximation and the extended Kalman filter on the large medical problem of predicting adverse outcome in pregnancy.

consistent win over the variational approach.

4. DISCUSSION

This paper shows that starting from the well known state space formulation one could derive sequential update algorithms for Bayesian estimation of the logistic regression model. With specific settings in the EKF formulation we achieve algorithms very similar to two other algorithms in the machien learning literature. Relaxing these settings we are able to perform better. Other interesting topics along these lines include more accurate characterisation of the probability densities. With nonlinear models one would expect to have multi modal distributions. Sequential sampling methods to achieve better representations are currently under study [4].

References

[1] Bishop, C.M. (1995), *Neural Networks for Pattern Recognition*, Oxford.

[2] Bar-Shalom, Y. & Fortman, T.E. (1988), *Tracking and Data Association*, Vol. 179 in Mathematics in Science and Engineering, Academic Press.

[3] De Freitas, G.F.J., Niranjan, M. & Gee, A.H. (1997), Hierarchical Kalman-Bayes models for Regularisation and Sequential Learning, `CUED/F-INFENG/TR.307`, Available in `http://www-svr.eng.cam.ac.uk/~jfgf`

[4] De Freitas, G.F.J., Niranjan, M., Gee, A.H. & Doucet, A. (1997), 'Sequential Monte Carlo Methods for Optimisation of Neural Network Models', `CUED/F-INFENG-TR 328`, Available via `http://www-svr.eng.cam.ac.uk/~jfgf`

[5] Jaakkola, T.S. & Jordan, M.I. (1996), 'A variational approach to Bayesian logistic regression models and their extensions', Available from `psyche.mit.edu`

[6] Jazwinski, A.H. (1970), *Stochastic Processes and Filtering Theory*, Vol. 64 in Mathematics in Science and Engineering, Academic Press.

[7] Kadirkamanathan, V. & Niranjan, M. (1993), 'A Function Estimation Approach to Sequential Learning with Neural Networks', *Neural Computation* **5**, pp. 954-975.

[8] Lovell, D.R., Scott, M.J.J., Niranjan, M, Prager, R.W., Dalton, K.J. B& Derom, R. (1997), 'On the use of expected attainable discrimination for feature selection in large scale medical risk prediction problems', Report `CUED/F-INFENG/TR.299`, Available from `http://www-svr.eng.cam.ac.uk/projects/qamc`.

[9] Niranjan, M. (1997), 'Sequential Tracking in Pricing Financial Options using Model Based and Neural Network Approaches', In Mozer, M.C., Jordan, M.I. & Petsche, T., Ed., *Advances in Neural Information Processing Systems* **9**, MIT Press, pp: 960-966.

[10] Penny, W.D. & Roberts, S.J. (1997), 'Bayesian Neural Networks for classification: how useful is the evidence framework?', Available from `http://www.ee.ic.ac.uk/hp/staff/sroberts.html`.

[11] Puskorius, G.V. & Feldkamp, L.A. (1994), 'Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks', *IEEE Transactions on Neural Networks*, 5(2), pp. 279-297.

[12] Quinlan, R. (1987), 'Simplifying decision trees', Int J Man-Machine Studies 27, pp. 221-234.

[13] Spiegelhalter, D. & Lauritzen, S. (1990), 'Sequential updating of conditional probabilities on directed graphical structures', Networks **20**: 579-605.