AN ITERATIVE SIGNAL DETECTION ALGORITHM BASED ON BAYESIAN BELIEF PROPAGATION IDEAS

I. Proudler*
QinetiQ Ltd.
St Andrews Road, Malvern
Worcestershire, WR14 2NJ

S. Roberts, S. Reece and I. Rezek
Dept. of Engineering Science
University of Oxford,
Parks Road, Oxford, OX1 3PJ

ABSTRACT
We investigate the construction of an iterative algorithm for signal detection based on Pearl’s Belief Propagation algorithm. Two main issues arise: the need to modify the graphical model of the problem in order to force the creation of an iterative algorithm, and the need to find suitable approximations to various probability distribution functions so as to allow an efficient implementation of the algorithm. The resulting algorithm is similar to a conventional deflation-based approach but it can make soft rather than hard decisions.

Index Terms—signal processing, iterative algorithms, Bayesian, belief propagation, Pearl’s algorithm.

1. INTRODUCTION
Parameter estimation is at the heart of much of modern adaptive digital signal processing. However most, if not all, practical estimation problems involve nonlinear functions when expressed in an ideal form. This causes problems because the resulting mathematics is too difficult for a closed form solution to be found. In order to develop an algorithm some sort of compromise in needed. One approach is to develop an iterative algorithm for example by splitting the problem into two or more sub-problems each of which can be solved given some estimates from the other sub-problems (deflation). The deficiency with this approach is that the resulting algorithm usually only converges to the correct solution if the initial guess is close by. However when it does converge one has a means to solve the difficult nonlinear problem in a very efficient, robust way.

A recent example of a highly successful iterative estimation algorithm is the Turbo Code decoding algorithm [1]. It has been shown recently that the Turbo Code decoding algorithm is an instantiation of Pearl’s Belief Propagation algorithm [2]. In the following, we investigate the application of Pearl’s algorithm to a fairly generic, signal detection problem. The motivation being to discover if a similarly powerful, iterative, detection algorithm can be created.

∗This work was sponsored by the Royal Academy of Engineering, the ES Domain of the UK MOD’s Research Acquisition Organisation, and QinetiQ.

2. SIGNAL DETECTION

2.1. Introduction
In this section we consider a problem that involves the detection of several signals, of known shape or ‘signature’, in noise. This is a common problem found, for example, in biomedical signal analysis, communications, radar and sonar.

The observed data is given by
\[ y = T_1(a_1, k_1) + T_2(a_2, k_2) + w(\sigma) \]
where \( w(\sigma) \) is Gaussian white noise, \( T_i(a_i, k_i) = a_i f(k_i) \) and \( a_i \) is the (unknown) signal amplitude, \( f() \) is the (known) signal response vector and \( k_i \) is the (unknown) position parameter. For simplicity we consider the case when the noise and the amplitude coefficients \( a_i \) are real. Note that because a signal can be located at any position in the data and with any amplitude, the priors for this problem are uninformative.

2.2. Pearl’s Algorithm
In a Bayesian inference problem one tries to estimate the posterior probability distribution of a set of variables given some measurements In general this is a complicated problem to solve but if the problem has structure then the posterior probability distribution will factor into simpler elements. The structure can be represented via a ‘graphica’ model: a directed graph where each node represents a variable and the links show the causal connections between the variables. Pearl’s algorithm calculates the marginal probabilities at the nodes based on ‘messages’ that are passed between the nodes [3].

2.3. Iterative Algorithm Design
The graphical model for the detection problem is shown in figure 1(a). The nodes labelled \( a_i \) and \( k_i \) represent the amplitude and position parameters, respectively, of the two signals. The nodes labelled \( T_1 \) and \( T_2 \) represent the contribution of each of the two signals to the observed data (i.e. \( a_i f(k_i) \)).
the node labelled $\lambda$ represents the noise $^1$ and the node labelled $Y$ represents the observed data. It can readily be seen from figure 1(a) that the graph is a tree. This means that Pearl’s algorithm for this problem is not iterative [3]. Furthermore it is relatively easy to see that the resulting algorithm is rather trivial: The belief in a given signal effectively reduces to a state-

1For arithmetic reasons we deal with the variable $\lambda = 1/\sigma^2$ instead of $\sigma$ - the former being known as the ‘precision’.

2The nodes $a$ and $b$ exchange the messages $a_n(b)$ and $\lambda_n(a)$ - see [3] for definitions.

3. APPROXIMATE MESSAGE DISTRIBUTIONS

3.1. Introduction

We desire to approximate various parameterised functions by other parameterised functions. In the Bayesian literature the ideal way to do this is to minimise the Kullback-Leibler divergence of the two functions. The KL divergence can be written:

$$KL(f(x|\theta), g(x|\phi)) = \int f(x|\theta) \ln \left( \frac{f(x|\theta)}{g(x|\phi)} \right) dx$$

where the parameters $\theta$ are given and $\phi$ are to be determined by the minimisation. Note that we do not need a good functional approximation but rather a good ‘information theoretic’ one that captures the relevant information.

3.2. Normal-Gamma Approximation to Normal x Gamma

We require to find a Normal-Gamma approximation to the product $\pi_y(T) \pi_y(\lambda)$ i.e. find $(\mu, L, \alpha, \beta)$ that minimises

$$KL(N(\mu|\tilde{\mu}, \tilde{L}), Ga(\lambda|\tilde{\alpha}, \tilde{\beta}), NG(\mu, \lambda|\mu, L, \alpha, \beta))$$

where $N(\cdot), Ga(\cdot)$ and $NG(\cdot)$ are Normal, Gamma and Normal-Gamma distributions respectively. It is not clear that this closed form solution to the KL minimisation problem exists. However if we approximate $E_\lambda \{ \ln (\lambda) \}$ by $\ln (E_\lambda \{ \lambda \})$, assume that the logarithm function is approximated constant, and align the peak of the Normal-Gamma with that of the Gamma, the KL divergence is approximately minimised when $\tilde{\mu} = \mu$, $\tilde{L} = \beta \lambda / \alpha$, $\alpha = r \beta - \frac{a}{2} + 1$ and $\beta = \exp \left( \left( \frac{a - 1}{r} \right) \right)$ where $r = \frac{a - 1}{b}$. The resulting approximation is good except in the region where $\lambda$ is small. The approximation captures the mean of $\mu$ correctly but the effect of the approximation error is to underestimate the precision of the contribution of a signal to the observed data at low SNRs.

Fig. 1. Graphical Models for Signal Detection Problem

The resulting approximation is good except in the region where $\lambda$ is small. The approximation captures the mean of $\mu$ correctly but the effect of the approximation error is to underestimate the precision of the contribution of a signal to the observed data at low SNRs.
3.3. Normal Approximation to Student t distribution

The message $\lambda_y(T)$ is a Student t distribution and we require a Normal approximation for it. Minimising the KL divergence results in an intractable set of equations. On the other hand, if we equate the mode of both distributions, in terms of position and height, and use Stirling’s approximation for the factorial function, we get a good approximation to the Student t distribution - this is not surprising given the form of the two distributions:

$$St(x \mid \mu, \lambda, \alpha) \approx N\left(x \mid \mu, \left(\frac{\lambda}{\alpha}\right) e^{-1} N\right)$$

3.4. Gamma Approximation to Integral of a Gamma

We require a Gamma approximation $\left(Ga\left(\lambda \mid \tilde{\alpha}, \tilde{\beta}\right)\right)$ for the message $\lambda_y(\lambda)$. Consider

$$\lambda_y(\lambda) = \int Ga\left(\lambda \mid \tilde{\alpha}, \tilde{\beta}\right) N\left(x \mid \mu, \mu, L\right) d\mu$$

Minimising the KL divergence directly does not lead to a tractable solution. Instead we set the position of the peak of $Ga\left(\lambda \mid \tilde{\alpha}, \tilde{\beta}\right)$ at the same point as the peak of $\lambda_y(\lambda)$, and assume that $E_x(\{f(\lambda)\}) \approx f\left(E_x(\{\lambda\})\right)$, the logarithm function and $\left(\frac{\lambda^{\tilde{\beta}+1}}{(L_0+L)\lambda^{\tilde{\beta}+1}}\right)$ are approximately constant, then we have that $\tilde{\alpha} = \frac{N\beta}{\alpha} \tilde{\beta}$ and $\beta = \left\|x - \left(\tilde{\mu} + \mu_0\right)\right\|^2$ where $\lambda_0 = \frac{N L}{(L_0+L)}$. If $L$ is large, $\lambda_y(\lambda)$ is very nearly a gamma distribution (see figure 2-a) as we get a good approximation. Otherwise, $\lambda_y(\lambda)$ rises from zero to a constant value (see figure 2-b) and represents little more than a lower bound on $\lambda$ and this is all that needs to be captured in the approximation. Note that there is no peak for the case $L = 20$ (see figure 2-b), so the position of the peak is limited to some (predetermined) maximum value.

4. SIGNATURE FUNCTION

Each $\lambda_y(T)$ message is represented as Normal distribution and therefore consists of a mean and precision. The mean is a representation of the contribution of this signal to the observed data. Ideally it should have the form $af(k)$ where $a$ and $k$ are as yet unknown. One way in which $a$ and $k$ can be estimated is by correlating the mean with the signature function. The position of the peak of the correlation function gives an estimate of $k$ and an estimate of $a$ can be found from its maximum value.

Having generated an estimate of $a$ and $k$, we now have to modify the $\lambda_y(T)$ message accordingly. This done via Bayes theorem. That is we consider the original $\lambda_y(T)$ message to be a likelihood and the estimate of the signal response based on $a$ and $k$ as the prior, then the updated message is the posterior. The prior ($p(T)$) can be modelled as a Normal distribution with mean $af(k)$. If the mean of the original $\lambda_y(T)$ message and $af(k)$ are dissimilar then we say that the precision of the prior is high and vice versa. However if the mean of the original $\lambda_y(T)$ was erroneous, giving the time series $af(k)$ a large precision would be wrong even if the correlation coefficient were close to unity. Hence we choose to set the precision of the prior equal to the correlation coefficient times the precision of the original $\lambda_y(T)$ message.

5. SIMULATION RESULTS

Some initial simulations were performed using the model described in section 2. As well as the new ‘Turbo’ algorithm a conventional deflationary algorithm was used. The data con-
Fig. 3. Data and Signal Responses

sisted of 100 time samples and the signal signature $f(k)$ was a sinc function truncated to 15 time samples. Three signals were placed close together at time instants 45, 50 and 55 with amplitudes 1.1, 1.8 and 1.3. The noise had a variance of unity. In order to add some realism, two inaccuracies were introduced. The algorithms had no knowledge of the number of signals present: Both algorithms assumed there were at most five signals. The algorithms also used an incorrect signature function: A sinc function was used where the width of the peak is narrower than the real signatures. In this case the formula for the precision of the signal signature prior (see section 4) was modified by weighting it by 0.5 to reflect the lack of confidence in the function being used.

Figure 3 shows the signal means estimated by the Turbo algorithm and a conventional algorithm, after 10 iterations, compared with the raw data. Both algorithms have detected five signals but there are two important differences. The conventional algorithm has placed all five signals in the peak of the data because the given signature function is narrower than the real one. In contrast, the Turbo algorithm has placed three signals in peak of the data and given them signatures that are wider than the given signature function - see figure 4(a). The remaining two, spurious, signals detected by the Turbo algorithm, unlike those for the conventional algorithm, have amplitudes comparable to the noise floor - see figure 4(b).

6. CONCLUSION

A straightforward application of Pearl’s algorithm to the signal detection problem leads to a non-iterative algorithm that is of little use. However it is possible to force the generation of an iterative algorithm by modifying the graphical model. Further more the use of numerical integration techniques has been avoided by the judicious use of approximations. The resulting algorithm is similar in complexity to a conventional deflation-based approach but the new algorithm can make soft rather than hard decisions and so is much more tolerant to incorrect assumptions about the problem.

7. REFERENCES


