

Dynamic programming versus graph cut algorithms for fitting non-parametric models to image data

C.A. Glasbey¹

¹ Biomathematics & Statistics Scotland, King's Buildings, Edinburgh EH9 3JZ, Scotland

Abstract: Dynamic programming and graph cut algorithms can, in some cases, find globally optimal fits of non-parametric models to image data, for restoration, segmentation and template matching. In this paper we compare conditions and results for the two methods, illustrated by restoration of a synthetic aperture radar (SAR) image.

Keywords: Image Restoration; Maximum a Posteriori Estimator; Markov Random Field; Penalised Likelihood; Synthetic Aperture Radar.

1 Introduction

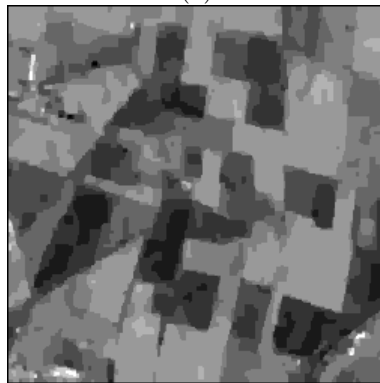
Non-parametric models are fit to image data for many reasons, including restoration, segmentation and template matching. For example, synthetic aperture radar (SAR) is a form of remote sensing in which data have a large noise component and therefore need smoothing, or restoring, to simplify interpretation. To illustrate, Figure 1(a) shows a log-transformed SAR image of an area near Thetford in England, obtained by plane in the Maestro-1 campaign. A pattern of agricultural fields can be discerned but, although we would expect the true signal to be approximately constant within each field, there is considerable speckle. SAR image restoration can be formulated as non-parametric smoothing:

$$\hat{\beta} = \arg \min_{\beta \in \mathcal{B}} \left\{ \sum_i (y_i - \beta_i)^2 + \sum_{\|i-j\|=1} \lambda |\beta_i - \beta_j| \right\}. \quad (1)$$

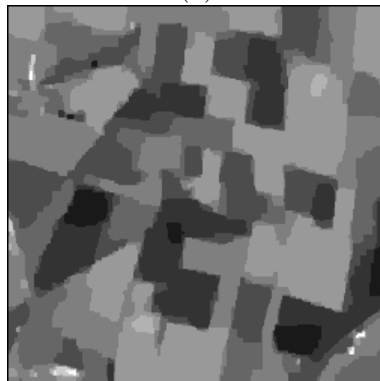
Here, y is the log-transformed SAR data, a $P = 250 \times 250$ array of scalars (rescaled to lie in the range 0 and 255) indexed by i , a 2-dimensional integer vector, and β is the scalar 2-dimensional array of restored values, an estimate of the true SAR signal, constrained to a finite set \mathcal{B} which, to speed-up computations, we restrict to a small set of values $\{65, 75, \dots, 165\}$. Summations are over all possible values of i and of i and j , and $\|\cdot\|$ denotes the Euclidean metric. So, we seek the least-squares restoration of y , subject to a penalty for lack of smoothness in β , specified by an absolute difference penalty with a first-order neighbourhood, and scaled by λ , a non-negative



(a)



(b)



(c)

FIGURE 1. Synthetic aperture radar (SAR) image: (a) log-transformed data downloadable from <ftp://ftp.bioss.sari.ac.uk/pub/chris/warping/>; (b) restoration using iterated dynamic programming (IDP); (c) restoration using graph cut algorithm.

constant whose magnitude determines the smoothness of the fit. By using this form of penalty, we impose smoothness on the restoration while tolerating step changes that we anticipate at agricultural field boundaries, and we set $\lambda = 50$, determined by eye to produce realistic restorations. Although the objective function in (1) is convex, this is a pathological optimisation problem for which gradient descent algorithms are unsuccessful (Kunsch, 1994).

More generally, image restoration, segmentation and template matching can often be formulated, using either a Bayesian or penalised likelihood framework, as

$$\hat{\beta} = \arg \min_{\beta \in \mathcal{B}} \left\{ \sum_i f(y, \beta_i) + \sum_{\|i-j\|=1} g(\|\beta_i - \beta_j\|) \right\}. \quad (2)$$

Here, β is the I -dimensional array of B -dimensional vectors that specifies the model fit, indexed by i , an I -dimensional integer vector, f is the measure of model fit to the data, derived from the negative log-likelihood of y , and g is either an empirical term penalising lack of smoothness in β , or the log-prior of β , a Markov Random Field (MRF) with first-order neighbourhood. We restrict β to a finite set $\mathcal{B} = \{1, 2, \dots, n\}^B$, though this is not overly restrictive since any desired level of resolution can be achieved by setting n sufficiently large and rescaling y . There are many choices for f and g , depending on the application. For image restoration, one option is $f = (y_i - \beta_i)^2$ as in (1), whereas for warping, or matching, to template μ , $f = (y_i - \mu_{i+\beta_i})^2$, and for segmentation we could use $f = y_{\beta_i}^*$, where y^* denotes transformed data (Glasbey and Young, 2002). The penalty term may be convex, such as $g(z) = \lambda z^2$ or $\lambda|z|$, as used in (1), or non-convex, such as the indicator function.

Solving (1) or (2) is computationally challenging for two reasons: image data sets are large and such objective functions are prone to local optima. However, if certain conditions apply, dynamic programming and graph cut algorithms can be used, and they are among that rare class of optimisation algorithms, those that are both fast and global!

2 Algorithms

Dynamic programming (DP) is an elegant method for finding the global solution to (2), but only if $I = 1$ (i.e. β is a one-dimensional array), using a sequential algorithm. Cases include 1-D image warping, also termed dynamic time warping, and finding 1-D boundaries to segment 2-D images, by finding a path between opposite sides of an image, with β_i specifying the row location of the boundary in column i (Glasbey and Young, 2002).

DP cannot be used to restore the SAR image by solving (1) because $I = 2$. However, Glasbey (2009) proposed generalisations of DP to solve higher

dimensional problems, but without the guarantee of global optimality; the simplest being iterated dynamic programming (IDP). IDP can be initialised by applying DP separately to each image column, and subsequently DP is applied alternately to all rows and columns, taking into account neighbouring values of β . After 26 iterations, which took 1.2sec of CPU time on a single core of a 3.2Ghz AMD Opteron processor, β converged to an approximation to the maximum *a posteriori* (MAP) or maximum penalised likelihood estimator shown in Figure 1(b), with a minimised value of the objective function of $388P$.

Graph cut (GC) algorithms can also be used to find the global solution to (2), provided $B = 1$ (i.e. β is a scalar array) and g is a convex function. GC reformulates the optimisation problem as finding the maximum flow through a network from a source to a sink, or equivalently, the minimum-cost subset of edges which disconnect the source from the sink. We create an $I + 1$ -dimensional array of $P \times (n + 1)$ nodes, indexed by (i, β) , with edges and directional flow capacities specified by:

$$\begin{aligned} C\{\text{source} \rightarrow (i, 1)\} &= \infty \\ C\{(i, n + 1) \rightarrow \text{sink}\} &= \infty \\ C\{(i, \beta) \rightarrow (i, \beta + 1)\} &= f(y, \beta) \\ C\{(i, \beta) \rightarrow (j, \beta) \mid \|i - j\| = 1\} &= g(1) \\ C\{(i, \beta) \rightarrow (j, \beta - k) \mid \|i - j\| = 1\} &= \{g(k + 1) - 2g(k) + g(k - 1)\} \text{ for } k \geq 1, \end{aligned}$$

where, without loss of generality we assume that $g(0) = 0$. The original, Ford-Fulkerson algorithm solves the flow problem by repeatedly finding a path from source to sink with spare capacity and maximising flow along this path. Grieg et al (1989) were the first to use GC in image analysis, to restore binary images. More recently, Boykov and co-workers (see, for example, Boykov and Kolmogorov, 2004) have extended GC to a broader class of image problems, and shown that a different type of GC algorithm, termed ‘push-relabel’, is more effective for these applications.

SAR image restoration (1) satisfies the conditions for GC. Further, because $g(z) = \lambda|z|$, the last, large set of edges is not needed, though there are still 750,000 nodes and 4.6 million edges. Boykov’s implementation of the GC algorithm (available at <http://vision.csd.uwo.ca/code/>) took 1.1sec of CPU time to find the globally optimal result. However, there are many alternative GC algorithms, and speed can vary enormously. For example, another public domain implementation, NETFLO (Nijenhuis and Wilf, 1978), using a Ford-Fulkerson type algorithm, took 10^3 more time (21 minutes) to reach the same solution. The MAP or maximum penalised likelihood estimator is shown in Figure 1(c), with a minimised value of the objective function of $374P$. We see that the agricultural fields are more clearly identified than in Figure 1(b), presumably because we have the global optimum rather than simply a local one.

3 Discussion

DP and GC can each fit non-parametric models to image data by finding the global solution to (2) provided certain conditions apply. For DP, the condition is $I = 1$, whereas for GC, $B = 1$ and g must be convex. As we have seen, GC can restore 2-D images, or any higher dimension, for convex choices of g , whereas DP can only restore 1-D images, and for higher dimensions we have to resort to approximations such as IDP, though they have the advantage of permitting non-convex g . Note that, for the special case in (1), Kovac and Smith (2009) have shown that a taut-string type algorithm can also find the global optimum. In unpublished work, we have also compared IDP and GC for finding 2-D surfaces to segment 3-D images (i.e. $B = 1, I = 2$), which DP cannot achieve. However, if instead the model required us to find a 1-D path across a 3-D image ($B = 2, I = 1$) then DP could do it but GC would not be able to. DP and GC can both be used in 1-D warping ($B = 1, I = 1$), but only GC can perform so-called $1\frac{1}{2}$ -D warping, which occurs, for example, when matching stereo image pairs, where no warping is needed between rows (i.e. $B = 1, I = 2$). However, neither DP nor GC can perform 2-D or 3-D warping.

Overall, GC is a much more complicated algorithm than DP, and its many variants can have an enormous range of performance speeds. DP is easier to generalise, for example to MRFs with higher-order neighbourhoods and to find local optima using IDP even when DP cannot be used. GC can also be adapted, to find local optima when conditions for global optimality are not satisfied, but then other competitors exist, such as ‘loopy belief propagation’ and ‘tree-reweighted message passing’ (Szeliski et al, 2008).

Acknowledgments: I am grateful to my colleague Alec Mann for assistance in using Boykov’s C++ algorithm. The work was funded by the Scottish Government.

References

- Boykov, Y., and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1124-1137.
- Glasbey, C.A. (2009). Two-dimensional generalisations of dynamic programming for image analysis. *Statistics and Computing*, **19**, 49-56.
- Glasbey, C.A., and Young, M.J. (2002). Maximum *a posteriori* estimation of image boundaries by dynamic programming. *Applied Statistics*, **51**, 209-221.

- Greig, D.M., Porteous, B.T., and Seheult, A.H. (1989). Exact maximum *a posteriori* estimation for binary images. *Journal of the Royal Statistical Society, Series B*, **51**, 271-279.
- Kovac, A., and Smith, A.D.A.C. (2009). Regression on a graph. (Available on arXiv, at <http://arxiv.org/abs/0911.1928v1>)
- Kunsch, H.R. (1994). Robust priors for smoothing and image restoration. *Annals of the Institute of Statistical Mathematics*, **46**, 1-19.
- Nijenhuis, A., and Wilf, H.S. (1978). *Combinatorial Algorithms* (2nd edition). London: Academic Press
- Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2008). A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**, 1068-1080.