Lagrangian based methods for finding MAP solutions for MRF models

Geir Storvik Geir Dahl

Submitted to JASA

Abstract

Finding maximum a posteriori (MAP) solutions from noisy images based on a prior Markov Random Field (MRF) model is a huge computational task. In this paper we transform the computational problem into an integer linear programming (ILP) problem. We explore the use of Lagrange relaxation (LR) methods for solving the MAP problem. In particular three different algorithms based on LR are presented. All the methods are competitive alternatives to the commonly used simulation-based algorithms such as simulated annealing (SA). In all the examples (including both simulated and real images) that has been tested, the best method essentially finds a MAP solution in a small number of iterations. In addition, LR methods provide lower and upper bounds for the posterior, which makes it possible to evaluate the quality of solutions and to construct a stopping criterion for the algorithm.

The main idea in the Lagrangian relaxation method is to decompose the original complex problem into several simpler subproblems. Each of these subproblems can be solved fast due to their simple structure. The dependence between the subproblems is incorporated through penalty terms that are modified throughout the computation. The first of our methods leads to separate subproblems for each pixel. In the second relaxation method, the subproblems are to optimize along vertical (or horizontal) lines of the image matrix. This means that optimization is only made for one direction only. Optimization along both horizontal and vertical lines can be combined by doubling the set of variables, resulting in the third relaxation method.

[0]Geir Storvik is Associate Professor at the Institute of Mathematics, University of Oslo, P.box 1053 Blindern, 0316 Oslo, NORWAY. He is also a senior scientist at the Norwegian Computing Center. Email: geirs@math.uio.no. Geir Dahl is Associate Professor at the Institute of Informatics, University of Oslo, P.box 1080 Blindern, 0316 Oslo, NORWAY. Email: geird@ifi.uio.no.

The authors are grateful to Gunnar Myhr (the MR-center in Trondheim, Oslo) and Arvid Lundervold (Section for Medical Image Analysis and Pattern Recognition, University of Bergen) for providing the MR-image and to Anne Schistad Solberg (Norwegian Computing Center) for providing the texture based SAR image.

1 Introduction

Segmentation of noisy images into a fixed number of classes is an important problem in image analysis. In order to get satisfactory results, the importance of taking spatial dependence into account has long been recognized. The Bayesian approach is to construct a prior model which describes the available knowledge about the image. A class of priors that has been of particular interest is the Markov Random Fields (Besag 1974, Geman & Geman 1984, Besag 1986). The Bayesian paradigm is to base all inference on the posterior distribution, and one possible estimate is the mode of the posterior, i.e. the maximum a posteriori (MAP) solution. Since the number of possible solutions is exponential in the number of pixels, finding a MAP solution is a great computational challenge.

In Geman & Geman (1984) simulated annealing, based on Markov Chain Monte Carlo algorithms, was proposed for finding a MAP solution. Pixel-wise updating based on the Gibbs sampler or the Metropolis-Hastings algorithm was originally used, but later on other versions, such as the line-wise updating (Qian & Titterington 1989) and the Swendsen-Wang algorithm (Swendsen & Wang 1987) allow larger regions to be updated at each step of the algorithm. These modifications can in some situations speed up the convergence rate. There are however still weaknesses with this approach: (i) Convergence results for SA are not satisfactory, and (ii) evaluation of a suboptimal solution (i.e. deviation from the optimal value) is difficult (or even impossible).

Some attempts have been made using other optimization techniques for solving the MAP problem. Besag (1986) introduced the ICM algorithm, which from a starting solution (usually a non-contextual classification solution) makes pixel-wise updating as long as an increase in the posterior probability is obtained. Only convergence to a local optimum is ensured with the ICM-algorithm. Greig, Porteous & Seheult (1989) demonstrated how an exact MAP solution can be found in the case of a two-class problem. The problem can then be formulated as a shortest path problem for which many different algorithms exist, the best being of order proportional to the square of the number of pixels involved, see Ahuja, Magnanti & Orlin (1993). An attempt to extend this idea to three or more classes was presented in Ferrari, Frigessi & Gonzaga De SÁ (1995). Their approach is however restricted to assuming a Bernoulli field for the noise. Further, only an approximative solution is obtained in that a (sometimes large) fraction of the image remains unclassified.

In this paper we present an approach based on Lagrangian relaxation (LR) for finding a MAP solution. The foundation is to formulate the problem as an integer linear programming (ILP) problem with all variables being 0-1. Such problems are in general NP-hard, but in some situations it is possible to construct efficient algorithms tuned towards the specific problem of interest (see Nemhauser & Wolsey (1988) and Ahuja et al. (1993) for a general discussion on ILP problems). Lagrangian relaxation is a general method that has led to good practical algorithms for a number of large scale ILP problems, making it a natural method to consider. The Lagrangian relaxation algorithm is an iterative procedure with the following important characteristics:

1. Lagrangian relaxation transforms the MAP problem to minimization of an upper bound for a MAP value. The function to minimize is convex (although nondifferentiable), making the optimization problem considerably easier.

2. A lower bound can also be found, so the quality of a given solution can be evaluated and stopping criteria for the algorithm can be constructed.

The upper bound will in general be strictly larger than the MAP value. The lower bounds are obtained from solutions that are candidates for MAP solutions, typically giving high (good) values when the upper bounds are small.

There may be a gap between the upper and lower bounds, and the size of this gap depends on the choice of relaxation. We will consider three such choices, resulting in three different algorithms. Experiments on simulated and real examples show that the three LR algorithm all give reasonable approximations to a MAP solution. The best LR algorithm essentially finds a MAP solution after a relatively small number of iterations.

The outline for the rest of the paper is as follows: In Section 2 we describe the Markov Random Field model and the data model that will be assumed. Further, a short review of Lagrangian relaxation techniques for solving ILP problems is given. In Section 3 we describe how Lagrangian relaxation can be applied for finding a MAP solution. First, the problem is transformed into an ILP problem. Thereafter, we discuss different relaxations of interest. Experiments on simulated and real images are given in Section 4 before the final summary and discussion is given in Section 5.

2 Background

2.1 Markov Random Field models

Assume an image contains n pixels, each pixel belonging to a set of classes $\mathcal{K} = \{1, ..., n_k\}$. Let c_i be the class of pixel $i \in \mathcal{I}$ where $\mathcal{I}$ is the set of pixels. The image $\boldsymbol{c} = (c_i : i \in \mathcal{I})$ is not directly observed, but can be estimated from a degraded image $\boldsymbol{z} = (z_i : i \in \mathcal{I})$. The connection between $\boldsymbol{z}$ and $\boldsymbol{c}$ is given through the conditional distribution $f(\boldsymbol{z}|\boldsymbol{c})$ for $\boldsymbol{z}$ given $\boldsymbol{c}$. We will assume conditional independence between pixels, so that

$$f(\boldsymbol{z}|\boldsymbol{c}) = \prod_{i \in \mathcal{I}} f(z_i|c_i). \tag{2.1}$$

In many situations some prior information about the true image $\boldsymbol{c}$ is available. The most simple (and most popular) model used is the Markov Random Field model (MRF), which build in smoothness properties in the image. In its simplest form it is given by

$$\pi(\boldsymbol{c}) = \frac{1}{Z} \exp\{\sum_{i \in \mathcal{I}} \alpha_{c_i} + \sum_{\{i,j\} \in \mathcal{N}} \beta I(c_i = c_j)\} \tag{2.2}$$

where $\{\alpha_k\}$ defines the prior probabilities for the different classes, $\beta > 0$ is a parameter giving the degree of smoothness, $\mathcal{N}$ is the set of all first order neighbor pairs and Z is a normalization constant making $\pi(\cdot)$ a proper distribution. $I(A)$ is the indicator function for an event A to occur.

A generalization of (2.2) would be to let β depend on the classes c_i and c_j, or even on the locations i and j. Both these extensions can be implemented with minor modifications in the methods to follow. However, to keep the notation simple, we shall consider model (2.2) only. The *posterior* distribution for $\boldsymbol{c}$ given $\boldsymbol{z}$ can be found through Bayes formulae:

$$\pi(\boldsymbol{c}|\boldsymbol{z}) \propto \pi(\boldsymbol{c}) f(\boldsymbol{z}|\boldsymbol{c}).$$

The Bayesian paradigm is to base all inference on this posterior distribution. In particular, an estimate of $\boldsymbol{c}$ is a maximum a posteriori (MAP) solution which is given by

$$\hat{c} = \underset{\boldsymbol{c}}{\operatorname{argmax}}\, \pi(\boldsymbol{c}|\boldsymbol{z}) = \underset{\boldsymbol{c}}{\operatorname{argmax}}\, \pi(\boldsymbol{c}) f(\boldsymbol{z}|\boldsymbol{c}). \tag{MAP}$$

Note that in general there may not be a *unique* MAP solution. We will consider the problem of finding one such solution.

2.2 Lagrangian relaxation

In this section we give a brief introduction to Lagrangian relaxation (LR) as applied to a general 0–1 integer linear programming (ILP) problem

$$\begin{array}{ll} \max & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \\ \text{(i)} & \boldsymbol{A}^1\boldsymbol{x} \leq \boldsymbol{b}^1; \\ \text{(ii)} & \boldsymbol{A}^2\boldsymbol{x} \leq \boldsymbol{b}^2; \\ \text{(iii)} & \boldsymbol{x} \in \{0,1\}^n. \end{array} \tag{2.3}$$

where $\boldsymbol{A}^1, \boldsymbol{A}^2$ are matrices and $\boldsymbol{b}^1, \boldsymbol{b}^2$ and $\boldsymbol{c}$ are column vectors of some (finite) dimension. Note that this form also includes equalities as each equality may be written as two inequalities.

In integer and combinatorial optimization one is concerned with mathematical and algorithmic questions concerning problems of the form (2.3). In fact, this is a very active research area in mathematical optimization. As (2.3) is hard to solve in general (it is *NP*-hard, see Garey & Johnson (1979)) it is important to exploit the specific structure of a given problem in order to develop efficient algorithms. For general techniques such as cutting plane algorithms, relaxation techniques, branch and bound methods and different heuristics, we refer to the comprehensive treatment in Nemhauser & Wolsey (1988), see also Ahuja et al. (1993) for a treatment on LR.

Lagrangian relaxation may be viewed as an extension of the classical method of Lagrangian multipliers used to minimize a function subject to equality constraints. The extension is to handle inequalities: one relaxes some inequalities and includes these in the objective function such that a (linear) penalty occurs for solutions that violate any of the relaxed constraints. More specifically, consider the following problem which we denote by $LR(\boldsymbol{\lambda})$

$$\begin{array}{ll} \max & \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{\lambda}^T(\boldsymbol{A}^2\boldsymbol{x} - \boldsymbol{b}^2) \\ \text{subject to} & \\ \text{(i)} & \boldsymbol{A}^1\boldsymbol{x} \leq \boldsymbol{b}^1; \\ \text{(ii)} & \boldsymbol{x} \in \{0,1\}^n. \end{array} \tag{2.4}$$

where the vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ consists of a nonnegative *Lagrangian multiplier* ("penalty") λ_i associated with the ith inequality in $\boldsymbol{A}^2\boldsymbol{x} \leq \boldsymbol{b}^2$. Thus, in $LR(\boldsymbol{\lambda})$ a feasible point $\bar{\boldsymbol{x}}$ may violate a constraint $\boldsymbol{a}_i^T\boldsymbol{x} \leq \boldsymbol{b}_i$ in $\boldsymbol{A}^2\boldsymbol{x} \leq \boldsymbol{b}^2$ but this reduces the objective function by the amount of $\lambda_i(\boldsymbol{a}_i^T\bar{\boldsymbol{x}} - \boldsymbol{b}_i)$. The problem $LR(\boldsymbol{\lambda})$ is called the *Lagrangian subproblem* w.r.t. the constraints $\boldsymbol{A}^2\boldsymbol{x} \leq \boldsymbol{b}^2$. The main idea in Lagrangian relaxation is to relax "complicating" constraints such that the problem $LR(\boldsymbol{\lambda})$ is easier to solve.

Let P denote the problem (2.3) and $v(P)$ its optimal value. From the mentioned penalty of violating constraints it follows that

$$v(P) \leq v(LR(\boldsymbol{\lambda})).$$

Since this holds for all $\boldsymbol{\lambda} \geq 0$, the best upper bound obtained in this way is given by solving the so-called *Lagrangian dual problem* (LD) (w.r.t. $\boldsymbol{A}^2\boldsymbol{x} \leq \boldsymbol{b}^2$):

$$\min\{v(LR(\boldsymbol{\lambda})) : \boldsymbol{\lambda} \geq 0\} \tag{2.5}$$

and we get the important inequalities

$$v(P) \leq v(LD) \leq v(LR(\boldsymbol{\lambda})) \text{ for all } \boldsymbol{\lambda} \geq 0. \tag{2.6}$$

Note that if, for some $\boldsymbol{\lambda} \geq 0$, one finds an optimal solution $\boldsymbol{x}$ of $LR(\boldsymbol{\lambda})$ which also satisfies the relaxed constraints $\boldsymbol{A}^2\boldsymbol{x} \leq \boldsymbol{b}^2$, then we can conclude that $\boldsymbol{x}$ is in fact optimal in the original problem P. This follows from (2.6). Frequently, in applications of Lagrangian relaxation, one also develops heuristics for finding (hopefully) good feasible solutions in P. Each such heuristic solution provides a lower bound on the optimal value $v(P)$. Thus, one has both upper and lower bounds on the optimal value, and therefore also a performance guarantee on the quality of the best solution found. This is a main strength of combining relaxation techniques with heuristics.

The Lagrangian dual problem may be viewed as a nondifferentiable convex minimization problem as $v(LR(\boldsymbol{\lambda}))$ is a piecewise linear and convex function (it is the pointwise maximum of a finite number of affine functions). Algorithmically one tries to solve the Lagrangian dual problem by some kind of multiplier adjustment technique. The basic principle is to adjust the multiplier vector according to the current optimal solution $\boldsymbol{x}_s$. If $\boldsymbol{x}_s$ violates a constraint, the corresponding penalty (multiplier) is increased, but if $\boldsymbol{x}_s$ satisfies the constraint, the penalty is decreased. Different ideas are used for deciding how much these adjustments should be, and for this good strategies are problem dependent. A general technique, called the *subgradient method* (see Nemhauser & Wolsey (1988)) is to update $\boldsymbol{\lambda}$ by

$$\boldsymbol{\lambda}^{s+1} = \max\{0, \boldsymbol{\lambda}^s - \theta_s(\boldsymbol{b}^2 - \boldsymbol{A}^2\boldsymbol{x}_s)\}. \tag{2.7}$$

Here $\{\theta_s\}$ is a sequence of positive step lengths satisfying

$$\lim_{s\to\infty} \theta_s = 0, \quad \sum_{u=1}^{\infty} \theta_s = \infty$$

which assures that $\boldsymbol{\lambda}^s$ converges to an optimal solution of the Lagrangian dual problem. In particular, one may choose $\theta_s = 1/s$. Another choice which has shown to perform better in practice is

$$\theta_s = \eta_s \frac{\boldsymbol{c}^T\boldsymbol{x}_s - \boldsymbol{\lambda}_s^T(\boldsymbol{A}^2\boldsymbol{x}_s - \boldsymbol{b}^2) - L_s^*}{\|\boldsymbol{A}^2\boldsymbol{x}_s - \boldsymbol{b}^2\|^2} \tag{2.8}$$

where L_s^* is a lower bound obtained after s iterations of the algorithm and $\boldsymbol{x}_s$ is an optimal solution of $LR(\boldsymbol{\lambda}_s)$. η_s is a scalar which has to be chosen (strictly) between 0 and 2. A popular choice is to let $\eta_0 = 2$ and then reduce η_s by a factor of 2 whenever there has been no improvement in a specified number of iterations.

A problem with the updating of the Lagrangian parameters given by (2.7) is that the direction of the subgradient $\boldsymbol{b}^2 - \boldsymbol{A}^2\boldsymbol{x}_s$ may change dramatically from one iteration to another. A general suggestion (Holmberg 1996) is to be somewhat conservative by using a (weighted) average of the subgradients from current *and* previous iterations when updating $\boldsymbol{\lambda}$. For instance, one may give weight 0.7 on the current iteration and 0.3 on the previous one.

Note that if θ_s is updated according to (2.8), the lower bound L_s^* obtained will influence the steplengths used. In particular, by calling good heuristics one may improve L_s^* which may give better steplengths in the sense of faster convergence.

In practice, it may not be straightforward to find efficient algorithms based on Lagrangian relaxation. First, there may be many different ways of modeling the problem (i.e., choosing variables and constraints as in (2.3)), and the bounds obtained from different formulations may vary significantly. Secondly, for a given model, it is not obvious which constraints that should be relaxed. Loosely speaking, if the relaxed problem is too simple, the bound $v(LD)$ is typically not good.

The discussion above has been focused on inequality constraints, but equality constraints may be relaxed simultaneously. The only difference being that the corresponding Lagrangian multipliers are in this case no longer restricted to be nonnegative, but may vary over the whole real line.

Similar methods apply to the more general case where $\boldsymbol{x}$ is integral, but this will not be of interest here. We will for simplicity refer to the 0–1 linear programming problem as the ILP problem although this usually refers to the more general situation.

3 Methods

The main problem in finding a MAP solution is the two-dimensional dependence between pixels. For one-dimensional situations, the maximization can be seen as a shortest path problem and recursive algorithms based on dynamic programming, such as the Viterbi algorithm (Viterbi 1967), are available. For two-dimensional problems effective algorithms are unlikely to be found due to the spatial dependence. However, relaxations removing the spatial dependence is possible to construct. We will in this section consider different types of relaxations, resulting in new algorithms for the MAP problem.

The foundation of our approach is to transform the problem to an ILP problem. This makes the general theory reviewed in 2.2 directly applicable. We will in this paper concentrate on Lagrangian relaxation techniques. We remark however that many other types of methods do exist for solving ILP problems, which in principle also are applicable.

3.1 Transforming the MAP problem to an integer linear problem

Define

$$U(\boldsymbol{c}) = \sum_{i\in\mathcal{I}} \log f(z_i|c_i) + \sum_{i\in\mathcal{I}} \alpha_{c_i} + \sum_{\{i,j\}\in\mathcal{N}} \beta I(c_i = c_j) \tag{3.1}$$

which is the logarithm of the posterior distribution (neglecting a constant term), so maximizing the posterior is equivalent to maximizing U. The latter is usually referred to as (minus) the energy function. We can rewrite equation (3.1) to

$$U(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}} \gamma_{i,k}x_{i,k} + \sum_{\{i,j\}\in\mathcal{N}} \beta y_{i,j} \tag{3.2}$$

where

$$\begin{array}{llll} \text{(i)} & \gamma_{i,k} = \alpha_k + \log f(z_i|k), & i\in\mathcal{I}, k\in\mathcal{K}, \\ \text{(ii)} & x_{i,k} = I(c_i = k), & i\in\mathcal{I}, k\in\mathcal{K}, \\ \text{(iii)} & y_{i,j} = I(c_i = c_j), & \{i,j\}\in\mathcal{N}, k\in\mathcal{K}. \end{array} \tag{3.3}$$

Here $y_{i,j}$ can be interpreted as a "bond" variable between pixels i and j, similar to the auxiliary variables introduced in the Swendsen-Wang algorithm (Swendsen & Wang 1987) (to simplify we write $y_{i,j}$ although we do not have any ordering between i and j, that is $y_{i,j}$ and $y_{j,i}$ denote the same variable). We see that we have transformed U to a linear function of $(\boldsymbol{x},\boldsymbol{y})$.

There are constraints on $(\boldsymbol{x},\boldsymbol{y})$. From the definition, the x's and y's are 0-1 variables. Further the x's have to sum to 1 at each pixel. The connection between x's and y's is given by

$$y_{i,j} = 1 - \max_{k\in\mathcal{K}} |x_{i,k} - x_{j,k}|, \quad \{i,j\}\in\mathcal{N}. \tag{3.4}$$

This is easy to see using the definition of $x_{i,k}$ and $y_{i,j}$.

We have now transformed the problem to an optimization problem with a linear objective function. The constraint (3.4) is however nonlinear and a reformulation is needed. The problem we will consider, also denoted by ILP is

$$\begin{array}{lll} \text{maximize} & U(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\gamma_{i,k}x_{i,k} + \sum_{\{i,j\}\in\mathcal{N}}\beta y_{i,j} & \\ \text{subject to} & & \\ & \text{(i)} \quad x_{i,k}\in\{0,1\}, y_{i,j}\in\{0,1\} & i\in\mathcal{I}, k\in\mathcal{K}, \{i,j\}\in\mathcal{N}; \\ & \text{(ii)} \quad \sum_{k\in\mathcal{K}} x_{i,k} = 1, & i\in\mathcal{I}, k\in\mathcal{K}; \\ & \text{(iii)} \quad y_{i,j} \le 1 - x_{i,k} + x_{j,k}, & \{i,j\}\in\mathcal{N}, k\in\mathcal{K}; \\ & \text{(iv)} \quad y_{i,j} \le 1 + x_{i,k} - x_{j,k}, & \{i,j\}\in\mathcal{N}, k\in\mathcal{K}. \end{array} \tag{3.5}$$

It is easy to see that (3.4) implies the inequalities (iii) and (iv) of (3.5). The inequalities force $y_{i,j}$ to be equal to zero if the neighbor pixels have different class memberships. On the other hand, if the class memberships are equal, the inequalities reduce to $y_{i,j} \leq 1$, and not to equality to 1 as it should be. However, if $\beta > 0$ and $(\boldsymbol{x}, \boldsymbol{y})$ is optimal for the problem above, $x_{i,k} = x_{j,k}$ for all k will *always* imply that $y_{i,j} = 1$ (due to $\beta > 0$ in the term $\beta y_{i,j}$ in $U(\boldsymbol{x}, \boldsymbol{y})$). An optimal value for the ILP problem is therefore equivalent to a MAP value of the original problem.

The number of unknown in the ILP problem is $nn_k + |\mathcal{N}|$. Further, (3.5) contains $2n_k|\mathcal{N}|$ inequalities. Here $|\mathcal{N}|$ is the number of neighbor pixels in the image ($\simeq 2n$). In addition we have the constraints given by (i)-(ii) in (3.5), but as we shall see, these constraints will be easy to handle for the algorithms to follow.

The ILP problem described above is still a hard problem to solve (except when $n_k = 2$). The next step is therefore to relax some of the constraints involved.

3.2 Relaxing spatial dependence

We first consider the Lagrangian relaxation obtained by relaxing all the inequalities that give dependencies between $\boldsymbol{x}$ and $\boldsymbol{y}$. The Lagrangian subproblem denoted $LR1(\boldsymbol{\lambda})$ then becomes

$$\begin{array}{ll} \text{maximize} & U(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) \\ \text{subject to} & \text{(i)-(ii) in (3.5)} \end{array} \qquad (3.6)$$

where

$$\begin{aligned} U(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}) = & \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \gamma_{i,k} x_{i,k} + \sum_{\{i,j\} \in \mathcal{N}} \beta y_{i,j} + \\ & \sum_{\{i,j\} \in \mathcal{N}} \sum_{k \in \mathcal{K}} [\lambda_{i,j,k,0}(1 - y_{i,j} - x_{i,k} + x_{j,k}) + \lambda_{i,j,k,1}(1 - y_{i,j} + x_{i,k} - x_{j,k})] \\ = & \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \tilde{\gamma}_{i,k} x_{i,k} + \sum_{\{i,j\} \in \mathcal{N}} [\beta - \sum_{k \in \mathcal{K}} (\lambda_{i,j,k,0} + \lambda_{i,j,k,1})] y_{i,j} + \sum_{\{i,j\} \in \mathcal{N}} \sum_{k \in \mathcal{K}} [\lambda_{i,j,k,0} + \lambda_{i,j,k,1}] \end{aligned}$$

and we have introduced

$$\begin{aligned} \tilde{\gamma}_{i,k} &= \gamma_{i,k} + \sum_{j:\{i,j\} \in \mathcal{N}} s_{i,j}(\lambda_{i,j,k,0} - \lambda_{i,j,k,1}), \\ s_{i,j} &= \begin{cases} 1 & \text{if } i < j; \\ -1 & \text{if } i > j. \end{cases} \end{aligned}$$

Here $i < j$ means that either the x- or the y-coordinate of pixel i is smaller than the one in y. The need for introducing $s_{i,j}$ is due to an implicit ordering between i and j in the Lagrangian multipliers $\lambda_{i,j,k,0}$ and $\lambda_{i,j,k,1}$. For each $\boldsymbol{\lambda}$ it is easy to maximize U subject to the constraints involved. Indeed, an optimal solution is given by

$$\begin{aligned} x_{i,k}(\boldsymbol{\lambda}) &= \begin{cases} 1 & \text{if } k = \text{argmax}_l \tilde{\gamma}_{i,l}, \\ 0 & \text{otherwise,} \end{cases} \\ y_{i,j}(\boldsymbol{\lambda}) &= \begin{cases} 1 & \text{if } \beta > \sum_{k \in \mathcal{K}} (\lambda_{i,j,k,0} + \lambda_{i,j,k,1}), \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \qquad (3.7)$$

for each $\{i, k\}$ and $\{i, j\}$. (In case of more than one possible k corresponding to the maximum of $\tilde{\gamma}_{i,l}$, an arbitrarily choice can be made). We let $\boldsymbol{x}(\boldsymbol{\lambda})$ and $\boldsymbol{y}(\boldsymbol{\lambda})$ denote the solution in (3.7) for a given $\boldsymbol{\lambda}$.

The posterior probability for class k at pixel i given z_i only is proportional to $\exp\{\gamma_{i,k}\}$. The introduction of the Lagrangian multipliers can be interpreted as a modification of the class-probabilities with respect to the class-values of its neighbors, which is very similar to the idea in pixel-wise Gibbs sampling.

This gives the procedure for optimization of U for a given $\boldsymbol{\lambda}$ value. The Lagrangian dual problem is to minimize $v(LR(\boldsymbol{\lambda}))$ for $\boldsymbol{\lambda} \geq 0$. This problem is solved using the subgradient procedure, and $\boldsymbol{\lambda}$ is updated by

$$\begin{aligned} \lambda^{s+1}_{i,j,k,0} &= \max\{0, \lambda^{s}_{i,j,k,0} - \theta_s(1 - y_{i,j} - x_{i,k} + x_{j,k})\}, \\ \lambda^{s+1}_{i,j,k,1} &= \max\{0, \lambda^{s}_{i,j,k,1} - \theta_s(1 - y_{i,j} + x_{i,k} - x_{j,k})\}. \end{aligned} \tag{3.8}$$

An optimal solution $(\boldsymbol{x}(\boldsymbol{\lambda}), \boldsymbol{y}(\boldsymbol{\lambda}))$ of $LR(\boldsymbol{\lambda})$ may not be feasible in ILP as it may violate (iii) or (iv) of (3.5). A heuristic for transforming such a solution to a feasible one can be constructed by ignoring the y-variables and define the solution as the one specified by the x's only. There is also a possibility of improving this estimate locally by performing some ICM iterations with the given current solution as a starting point (corresponding to choosing a better but more time-consuming heuristic).

3.3 Relaxing dependence in one direction

Optimization along a one-dimensional set of data is easy. In this section (and also the next) we will see how this can be utilized. The idea is to decompose the ILP problem into subproblems involving maximization along horizontal or vertical lines in the image.

Two neighbor pixels are either on the same horizontal or vertical line. Thus we partition $\mathcal{N}$ as follows:

$$\begin{aligned} \mathcal{N}^h &= \{\{i,j\} \in \mathcal{N} : \ i \text{ and } j \text{ are on the same horizontal line}\}, \\ \mathcal{N}^v &= \{\{i,j\} \in \mathcal{N} : \ i \text{ and } j \text{ are on the same vertical line}\}. \end{aligned}$$

We now relax those inequalities for which pixels $\{i,j\} \in \mathcal{N}^h$ (an alternative could be to make the relaxation on the set $\mathcal{N}^v$). The Lagrangian problem $LR2(\boldsymbol{\lambda})$ in this case becomes

$$\begin{array}{ll} \text{maximize} & \\ U(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\lambda}) = & \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\tilde{\gamma}_{i,k}x_{i,k} + \sum_{\{i,j\}\in\mathcal{N}^v}\beta y_{i,j} + \\ & \sum_{\{i,j\}\in\mathcal{N}^h}[\beta - \sum_{k\in\mathcal{K}}(\lambda_{i,j,k,0}+\lambda_{i,j,k,1})]y_{i,j} \\ \text{subject to} & \\ & \text{(i)-(ii) in (3.5), and} \\ & y_{i,j} \leq 1 - x_{i,k} + x_{j,k}, \quad \{i,j\} \in \mathcal{N}^v, k \in \mathcal{K}; \\ & y_{i,j} \leq 1 + x_{i,k} - x_{j,k}, \quad \{i,j\} \in \mathcal{N}^v, k \in \mathcal{K}. \end{array} \tag{3.9}$$

The weights $\tilde{\gamma}_{i,k}$ are in this case given by

$$\tilde{\gamma}_{i,k} = \gamma_{i,k} + \sum_{j:\{i,j\}\in\mathcal{N}^h} s_{i,j}(\lambda_{i,j,k,0} - \lambda_{i,j,k,1}), \tag{3.10}$$

while $s_{i,j}$ is defined as before. Note that there are no constraints on the $y_{i,j}$'s for $\{i,j\} \in \mathcal{N}^h$, so these variables can be maximized explicitly (by choosing $y_{i,j} = I(\beta \geq \sum_{k\in\mathcal{K}}(\lambda_{i,j,k,0} + \lambda_{i,j,k,1})))$. For the other variables, it is more complicated. By inspecting the structure in the inequalities, there are now only constraints between variables on the same vertical line of the image. Therefore we can maximize each line separately. In this case the posterior probability weights of a pixel is modified by factors depending on the classes for pixels at neighboring lines, similar to the line-wise Gibbs sampling proposed by Qian & Titterington (1989). Maximization along a line can be performed by the Viterbi algorithm and is described in detail in the appendix.

The Lagrangian multipliers $\boldsymbol{\lambda}$ may be updated exactly the same way as before (i.e. using (3.8)). Concerning heuristics for transforming $(\boldsymbol{x}, \boldsymbol{y})$ to a feasible solution, this can again be based on the x's alone.

Because we now relaxe a subset of the inequlities that were relaxed in LR1, we have

$$\min_{\boldsymbol{\lambda}} v(LR2(\boldsymbol{\lambda})) \leq \min_{\boldsymbol{\lambda}} v(LR1(\boldsymbol{\lambda}),$$

that is a better upper bound is obtained! Note however that more computation is needed at each iteration for LR2. If the difference in the upper bounds is small, the first method may therefore still be preferred.

3.4 Combining vertical and horizontal solution

In the previous section, we saw how the full spatial problem could be simplified to many one-dimensional ones along vertical (or alternatively horizontal) lines in the image matrix. In this section, we will see how horizontal and vertical maximization can be combined to a new formulation of the problem.

A general technique for obtaining better bounds by Lagrangian relaxation is so-called *cost splitting*. It was introduced in Jörnsten & Näsberg (1986) and applied to the generalized assignment problem, see also Nemhauser & Wolsey (1988). It consists in duplicating variables and the Lagrangian dual problem may be seen as finding an optimal distribution of the cost function between the two sets of variables. The trick in our case is to define two set of x-variables $\boldsymbol{x}^h$ and $\boldsymbol{x}^v$ corresponding to optimization along horizontal and vertical lines, respectively. In particular, consider the optimization problem

$$\begin{array}{lll}
\text{maximize} & & \\
\quad U(\boldsymbol{x},\boldsymbol{y}) = & \multicolumn{2}{l}{\frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\gamma_{i,k}x^h_{i,j} + \sum_{\{i,j\}\in\mathcal{N}^h}\beta y_{i,j} +} \\
 & \multicolumn{2}{l}{\frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\gamma_{i,k}x^v_{i,j} + \sum_{\{i,j\}\in\mathcal{N}^v}\beta y_{i,j}} \\
\text{subject to} & & \\
 & \text{(i)} \quad x^d_{i,k}\in\{0,1\}, y_{i,j}\in\{0,1\}, & i\in\mathcal{I}, k\in\mathcal{K}, d=h,v,\{i,j\}\in\mathcal{N}; \\
 & \text{(ii)} \quad \sum_{k\in\mathcal{K}} x^d_{i,k} = 1, & i\in\mathcal{I}, k\in\mathcal{K}, d=h,v; \\
 & \text{(iii)} \quad y_{i,j}\leq 1 - x^d_{i,k} + x^d_{j,k}, & \{i,j\}\in\mathcal{N}^d, k\in\mathcal{K}, d=h,v; \\
 & \text{(iv)} \quad y_{i,j}\leq 1 - x^d_{j,k} + x^d_{i,k}, & \{i,j\}\in\mathcal{N}^d, k\in\mathcal{K}, d=h,v; \\
 & \text{(v)} \quad x^h_{i,k} = x^v_{i,k}, & i\in\mathcal{I}, k\in\mathcal{K}.
\end{array} \tag{3.11}$$

Clearly this problem is equivalent to (3.5). Relaxing the last constraint, results in the Lagrangian problem $LR3(\boldsymbol{\lambda})$

$$\begin{array}{ll}
\text{maximize} & \\
\quad U(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\lambda}) = & \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\tilde{\gamma}^h_{i,k}x^h_{i,k} + \sum_{\{i,j\}\in\mathcal{N}^h}\beta y_{i,j} + \\
 & \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\tilde{\gamma}^v_{i,k}x^v_{i,k} + \sum_{\{i,j\}\in\mathcal{N}^v}\beta y_{i,j} \\
\text{subject to} & \\
 & \text{(i)-(iv) in (3.11);}
\end{array} \tag{3.12}$$

where

$$\begin{array}{ll}
\tilde{\gamma}^h_{i,k} & = \frac{1}{2}\gamma_{i,k} + \lambda_{i,k}, \\
\tilde{\gamma}^v_{i,k} & = \frac{1}{2}\gamma_{i,k} - \lambda_{i,k}.
\end{array} \tag{3.13}$$

Note that in this case, optimization can be made separately on $\{\boldsymbol{x}^h, \{y_{i,j}, \{i,j\} \in \mathcal{N}^h\}$ and $\{\boldsymbol{x}^v, \{y_{i,j}, \{i,j\} \in \mathcal{N}^v\}$. Further, when optimizing the first set of variables, there is no interaction between variables from different horizontal lines, making it possible to maximize each line separately. Similarly each vertical line can be maximized separately for the second set of variables. The weights $\gamma_{i,k}$ are in this case modified by moving the probabilities of class-memberships for the two set of variables towards each other. The maximization along horizontal and vertical lines can again be performed using the Viterbi algorithm described in the appendix.

Concerning the Lagrangian multipliers, these can in this case vary over the whole real line (because the constraints we have relaxed are equalities and not inequalities). The updating is performed by

$$\lambda_{i,k}^{s+1} = \lambda_{i,k}^{s} - \theta_s(x_{i,k}^h - x_{i,k}^v).$$

For a given $\boldsymbol{\lambda}$ a feasible solution can be defined by constructing a heuristic; for instance using only the $x_{i,j}^h$'s. More sophisticated transformations taking both the horizontal and vertical solutions into account is also possible. Finally, improvements can again be obtained by performing ICM-iterations or by some other techniques.

It should be noted that for the relaxation procedure described in this section, an actual implementation of the method can effectively be performed by working on the class-variables without the use of the indicator variables $x_{i,k}$ and $y_{i,j}$. This is so because all the maximization involved can be performed on the class-variables directly as described in the appendix. Such an implementation will particularly be important for reducing the memory requirements.

Finally we remark that the best upper bound using LR3 is at least as good as the one in LR2. A proof of this fact and other relations, will be presented in a forthcoming paper.

4 Experiments

In this section we explore the algorithms proposed in the previous section on different examples. We refer to the three Lagrangian relaxation methods described in 3.2, 3.3 and 3.4 by LR1, LR2 and LR3, respectively. For all the methods, the updating of the Lagrangian multipliers were performed as described in section 2.2, i.e. using (2.8) and the subgradients from both the current and previous iterations. The η_s's were modified whenever there was no improvement within two iterations. Further, ICM-updatings were performed at each iteration in order to improve the lower bounds. All computations have been performed on a SGI IRIX 5.3 work-station.

The Lagrangian relaxation methods are also compared to the performance of simulated annealing. Such a comparison is difficult due to the wide range of strategies possible for SA. The type of updating can be chosen to be pixel-wise, line-wise or on the Swendsen-Wang form (other types of updating is also possible). Further, the type of temperature schedule can be varied. Since geometric schedules are better than logarithmic ones in obtaining high MAP values in finite time, only such schedules are considered. Some experiments with the Swendsen-Wang algorithm showed that this method gives very slow convergence. The poor performance of this algorithm for conditional simulation has also been reported elsewhere, see for instance Gray (1993). The choice between line-wise and pixel-wise updating did not seem to give significant difference with respect to convergence. On the other hand, both the type of temperature schedule and the starting temperature are of high importance. For all the examples reported below, simulations were performed over a range of starting temperatures. Only pixel-wise updating was used. The run giving the highest posteriori value in the time range considered is in each case reported.

We will start with a simulated example. Figure 1 shows to the left a 80 × 80 class image simulated from a MRF model with $\beta = 0.7$ (the image has been smoothed a little bit to make it more realistic). The noisy image to the right has been obtained by simulating Gaussian variables

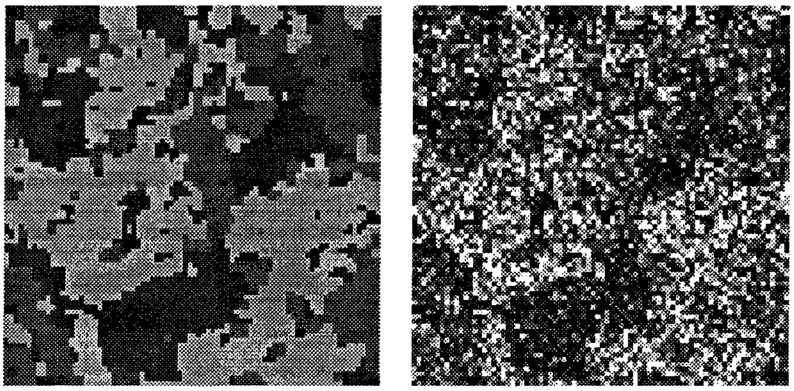

Figure 1: *Simulated class image (left) and noise corrupted image (right). The grey levels in the noisy image correspond to values in the range from 0 (black) to 255 (white).*

with mean equal to 64, 128 and 192, respectively for the three classes. The standard deviation was 64 for all the classes. The true parameter values were used in the computation.

The left plot in Figure 2 shows the upper and lower bounds for U as functions of computation time measured in seconds. For the LR1 method, the upper and lower bounds are given as the uppermost and lower-most curves in the figure. The large distance between these bounds indicate that the relaxed problem is too simple (further iterations did not improve the bounds). Turning to LR2 (the second upper and second lower curves), we see that an improvement is obtained compared to the LR1 method. Still there is some gap between the bounds. A further improvement is made when considering the method LR3. In this case the difference between the lower and upper bounds is negligible (less than 0.1). 57 iterations, taking 24.7 seconds was needed (further iterations gave no improvement). For comparison, the difference between the MAP value and the value obtained by ICM is 321.5, showing that a MAP solution is essentially found. In fact, only 9 pixels differ in the optimization along horizontal lines compared to optimization along vertical lines for the final value of $\boldsymbol{\lambda}$.

To the right in Figure 2, a comparison of the lower bounds from the LR methods are compared with values obtained from simulated annealing. We see that LR1 and LR2 are not capable of coping with the performance of simulated annealing, while LR3 is performing much better than simulated annealing.

One might ask for the practical importance of finding a MAP solution compared to the solution obtained by e.g. ICM. A MAP solution corresponds to the loss function giving zero loss if the true solution is found and loss equal to one otherwise. This is a very unrealistic loss-function. The other extreme which has been commonly applied is a loss function corresponding to minimizing the error rate. The main reason for choosing a MAP solution is that it gives more weight on the contextual structure. ICM is giving something between these extremes. In Figure 3, the classified images obtained by ICM (left) and LR3 (right) are shown. It is in this example clearly seen that the (nearly) MAP solution recognizes much more of the structures involved than is the case for the ICM solution.

Turn now to a real image. Figure 4 shows a PD-weighted MR image of the brain. The image is of size 256 × 256 and is to be segmented into five classes (fat, air/bones, connective tissue, cerebral fluid (CSF) and brain parenchyma). Gaussian distributions were used to describe the observations conditional on the classes. The parameters involved were obtained from a database consisting of a large set of MR images with partially known truth (provided from MR center in Trondheim, Norway). In addition to the PD-weighted image also a T1-weighted image was used for the segmentation. The smoothing parameter β was put to 0.7.

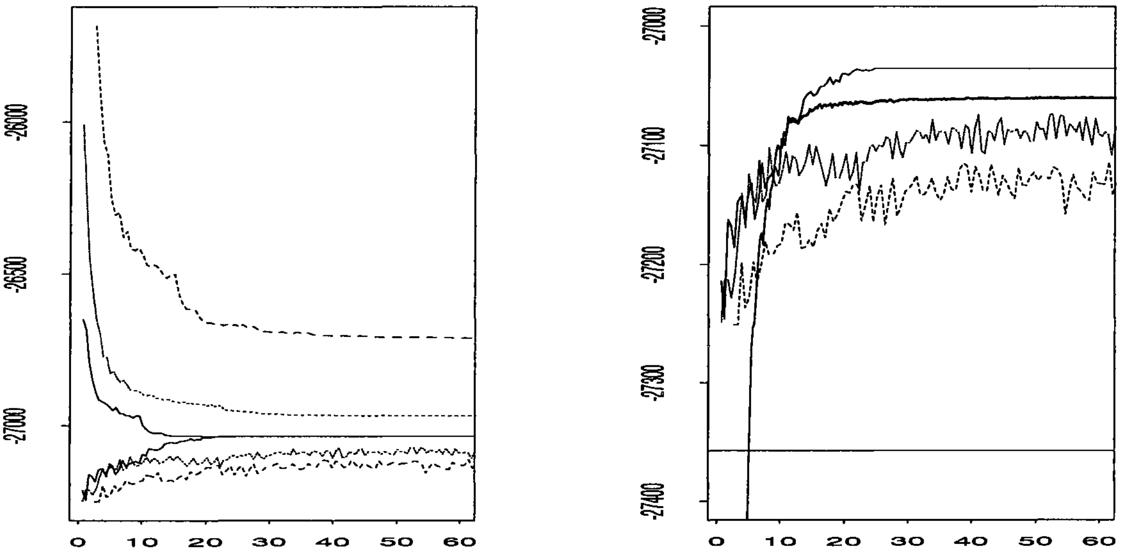

Figure 2: *To the left, upper and lower bounds for U for LR1 (dashed line) LR2 (dotted line) and LR3 (solid line) for the simulated image. To the right, lower bounds for LR1, LR2 and LR3 together with values obtained by simulated annealing (thick solid line). The result from the ICM algorithm is shown as a horizontal line with value -27357.5. For both plots, the x-axis gives the computation time in seconds.*

Again the three Lagrangian relaxation methods were tried out. The left plot in Figure 5 shows the upper and lower bounds for the energy function in this case. A very similar structure compared to the results obtained for the simulated image is seen. After 32 iterations (taking 285 seconds), the difference between the upper and lower bounds from the LR3 method is down to 0.15. In this case, the difference between the MAP value and value found by ICM is 212.5 which again demonstrate that the MAP solution is essentially found. Both the LR1 and LR2 methods performed worse than simulated annealing (right plot in Figure 5).

In Figure 4, the ICM solution (middle) and the best LR3 solution (right) are displayed. The images are in this case very similar, mainly because the MR images contain little noise. Note however the dark areas inside the white brain parenchyma which is classified to fat in the ICM solution. For a MAP solution these have been correctly classified to brain parenchyma.

The third example we consider is based on an ERS-1 SAR image over the area Kjeller/Lillestrøm, nearby Oslo, Norway. This image, which is of size 333 × 260 and aimed to be segmented to five classes (water, urban areas, forest, unplowed agricultural fields and plowed agricultural fields), was used for comparing texture features in Solberg & Jain (1996). We will consider the textures obtained from a multiplicative autoregressive random field (Frankot & Chellappa 1987).

One (out of five) texture images is shown at the left in Figure 6. A multivariate Gaussian model was assumed for the observations. The parameters were estimated from a training set consisting of regions in the image with known class-memberships. We used $\beta = 1.0$.

The results obtained with the Lagrangian relaxation techniques are shown in Figure 7. Similar to the other examples, the LR3 method is essentially finding a MAP solution. The difference between the upper and lower bounds is in this case 0.68 after 78 iterations, taking 912 seconds. Comparing this to the difference between the MAP value and the ICM value (4030.5), we again see that a MAP solution is in principle found. For the two solutions obtained from the LR3 method with the final λ value, only 10 pixels are differing. The two other methods are somewhat

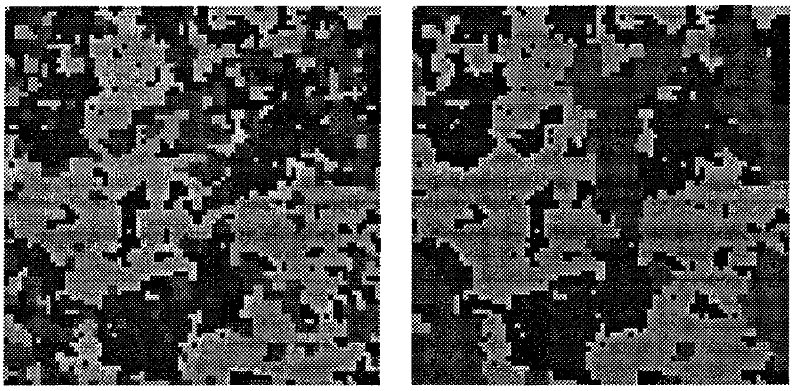

Figure 3: *Solutions obtained by ICM (left) and LR3 (right) from the simulated noise image.*

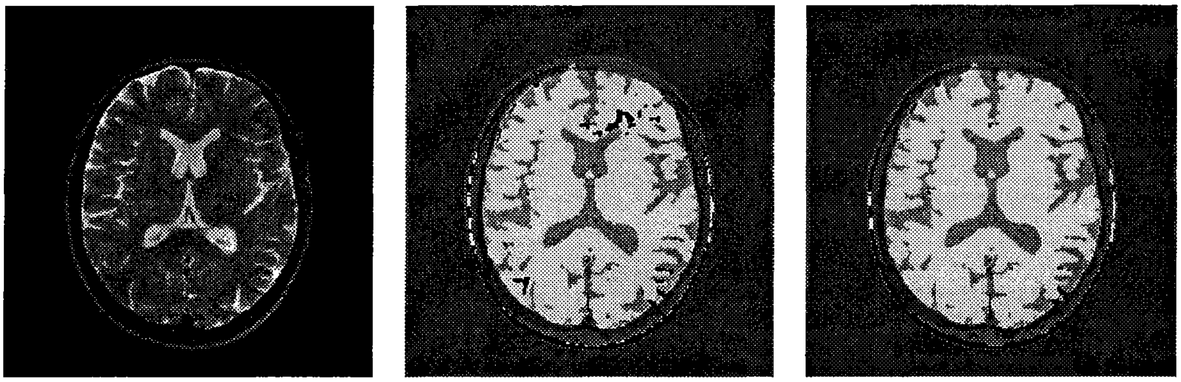

Figure 4: *To the left, T2-weighted Magnetic Resonance images of the Brain. The grey levels correspond to values in the range from 0 (black) to 255 (white). In the middle, solution obtained by ICM and to the right the solution obtained by LR3. The classes are fat, air/bone, connective tissue, CSF and brain parenchyma, displayed from dark to bright.*

poorer. Comparing to simulated annealing, we see again that a considerable improvement is made by the LR3 method, while LR2 perform better than SA.

In the middle and to the left of Figure 6 the solutions obtained by ICM and LR3 are displayed also for this example. The error rates on the training sets were almost identical for the two images (65.5 and 65.6%). Such measures is of course not the ones to be used for comparison, since a MAP solution is not optimal with respect to minimizing the error rate. Still, we see that in this case nothing is lost with respect to the error rate for using a MAP solution compared to ICM (which is more tuned towards error rate minimization), while at the same time a smoother and more realistic solution is obtained.

5 Summary and discussion

We have in this paper discussed the use of Lagrangian relaxation for constructing new algorithms in order to obtain (near-optimal) MAP solutions. The approach is based on transforming the problem into an integer linear programming problem with the objective function being a function of both indicator variables denoting if a pixel belong to a certain class and "bond" variables indicating if neighbor pixels have equal classes. Constraints are introduced due to the dependence between the class-membership variables and the bond-variables.

Different types of relaxations lead to three different methods. Among the three relaxations

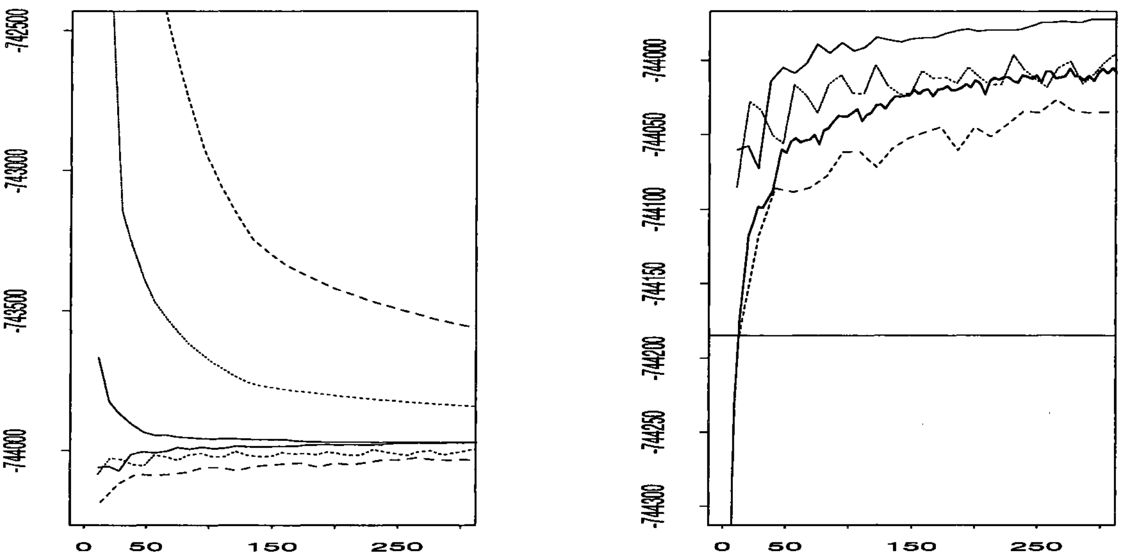

Figure 5: *To the left, upper and lower bounds for U for LR1 (dashed line), LR2 (dotted line) and LR3 (solid line) obtained on the MR images. To the right, lower bounds for LR1, LR2 and LR3 together with values obtained by simulated annealing (thick solid line). The result from the ICM algorithm is shown as a horizontal line with value -744185. For both plots, the x-axis gives the computation time in seconds.*

considered, a method based on combining optimization along horizontal and vertical lines in the image matrix turned out to give the best results. This is not surprising since much more information on the structure is built into the subproblems to be optimized than for the other methods. The ordering in the performance can also be shown theoretically.

The best Lagrangian relaxation algorithm performed much better than simulated annealing techniques in all examples considered (both simulated and real images). Further, the lower and upper bounds directly obtained by the algorithm were in all cases almost equal after a small number of iterations, implying that MAP solutions were essentially found. The existence of these bounds also makes it possible to evaluate a given solution.

The minimization of the Lagrangian dual (i.e. minimizing the lower bound) has been performed by applying a subgradient method. The method involves several details that may have significant influence on the convergence rate. No attempt has been made in order to "tune" these parameters and essentially general recommendations have been followed. By tuning the algorithm, further improvements should be possible to obtain.

The model we considered is the simplest first order Markov Random Field models in two dimensions. Extensions to models for which the spatial dependence depends on the classes involved and/or position can be incorporated with minor modifications. Similarly an extension to higher dimensions is possible. Extension to higher order dependence structures can be made by introducing additional variables defining the "bonds" between pixels at larger distances. The computational burden and the memory space needed will increase, but considering the performance of the methods for the simpler models, such extensions can be of importance.

The formulation of the MAP problem used in this work is not the only way to describe the problem as an ILP problem. Other choices are possible and can give rise to different algorithms with possibly better performance. This will be considered in forthcoming work.

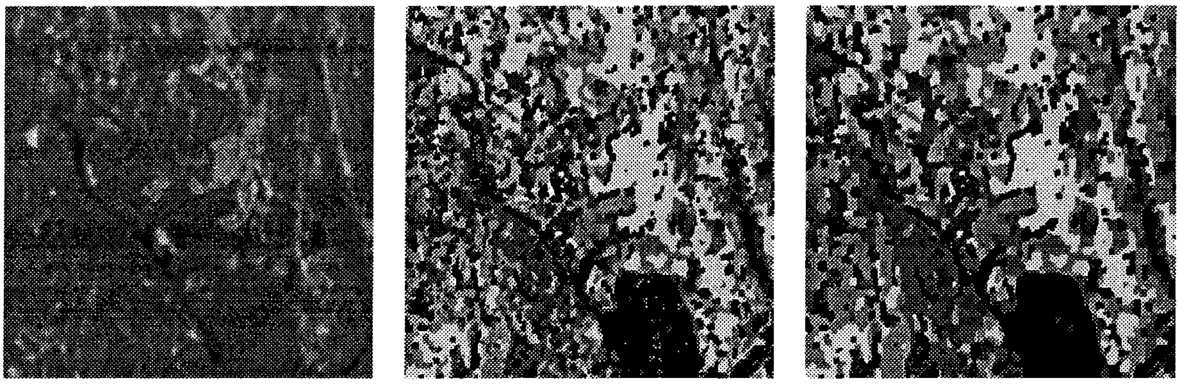

Figure 6: *To the left, a texture image computed at Norwegian Computing Center based on a SAR image from ESA/EURIMMGE/TSS/Spacetex. The grey levels correspond to values in the range from 0 (black) to 255 (white). In the middle, the solutions obtained by ICM and to the right, the solution obtained by LR3. The classes are water, urban areas, forest, agricultural fields (unplowed) and agricultural fields (plowed), displayed from dark to bright.*

A The Viterbi algorithm

We will in this section show how a optimal solution can be found along a one-dimensional series of pixels. Consider line r (assumed vertical for illustration). Denote the set of pixels along this line as $\mathcal{I}_r$. Denote further $\mathcal{N}_r^v$ as the set of neighbor pixels $\{i,j\}$ where both i and j belong to $\mathcal{I}_r$. Then the subproblem of $LR2(\boldsymbol{\lambda})$ (see (3.9)) corresponding to line r is

$$U_r(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\lambda}) = \sum_{i\in\mathcal{I}_r}\sum_{k\in\mathcal{K}}\tilde{\gamma}_{i,k}x_{i,k} + \sum_{\{i,j\}\in\mathcal{N}_r^v}\beta y_{i,j}$$

subject to

$$\begin{aligned}
&x_{i,k}, y_{i,j} \in \{0,1\}, && i \in \mathcal{I}_r, k \in \mathcal{K}, \{i,j\} \in \mathcal{N}_r^v;\\
&\sum_{k\in\mathcal{K}} x_{i,k} = 1, && i \in \mathcal{I}_r;\\
&y_{i,j} \leq 1 - x_{i,k} + x_{j,k}, \{i,j\} \in \mathcal{N}_r^v;\\
&y_{i,j} \leq 1 + x_{i,k} - x_{j,k}, \{i,j\} \in \mathcal{N}_r^v.
\end{aligned}$$

This can be rewritten to the maximization of

$$U_r(\boldsymbol{c}_r;\boldsymbol{\lambda}) = \sum_{i\in\mathcal{I}_r}\tilde{\gamma}_{i,c_i} + \sum_{\{i,j\}\in\mathcal{N}_r}\beta I(c_i = c_j) \tag{A.1}$$

with $\boldsymbol{c}_r = (c_i : i \in \mathcal{I})$ and $c_i \in \mathcal{K}$. Maximization of (A.1) corresponds to finding the most probable solution in an one-dimensional Markov chain problem. This can be formulated as a shortest path problem for which many algorithms exist. We will describe how a optimal sequence of classes can be found using the Viterbi algorithm. Denote the pixels in $\mathcal{I}_r$ by $(1,...,n_r)$. Then we can rewrite (A.1) to

$$U_{n_r}(\boldsymbol{c}_r) = \sum_{i=1}^{n_r}\tilde{\gamma}_{i,c_i} + \sum_{i=2}^{n_r}\beta I(c_i = c_{i-1}).$$

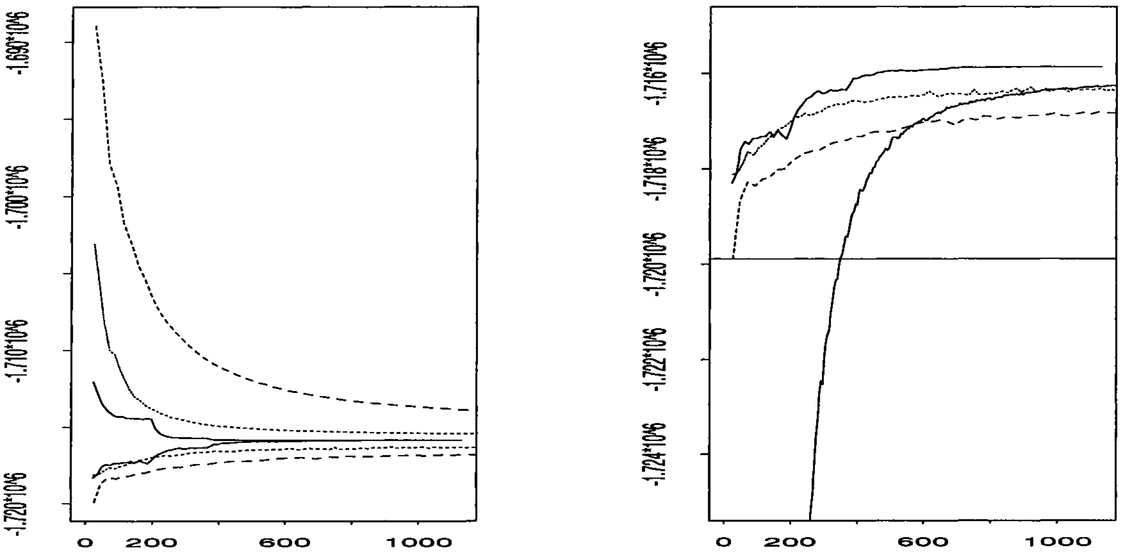

Figure 7: *To the left, upper and lower bounds for U for LR1 (dashed line), LR2 (dotted line) and LR3 (solid line) for the texture image. To the right, lower bounds for LR1, LR2 and LR3 together with values obtained by simulated annealing (thick solid line). The result from the ICM algorithm is shown as a horizontal line with value -1719895.86. For both plots, the x-axis gives the computation time in seconds.*

The algorithm is defined through the following four steps:

```
1. Initialization:  For $k = 1, ..., n_k$
                      $\psi_1(k) = 0,\ \delta_1(k) = \tilde{\gamma}_{1,k}$.
2. Recursion:       For $i = 2, ..., n_r$
                      For $k = 1, ..., n_k$
                        $\delta_i(k) = \max_{l \in \mathcal{K}}[\delta_{i-1}(l) + \beta I(k = l)] + \tilde{\gamma}_{i,k}$,
                        $\psi_i(k) = \text{argmax}_{l \in \mathcal{K}}[\delta_{i-1}(l) + \beta I(k = l)]$.
3. Termination:     $c^*_{n_r} = \text{argmax}_{k \in \mathcal{K}}[\delta_{n_r}(k)], \quad P^* = \delta_{n_r}(c^*_{n_r})$.
4. Backtracking:    For $i = n_r - 1, ..., 1$
                      $c^*_i = \psi_{i+1}(c^*_{i+1})$.
```

The sequence $(c^*_i, i = 1, ..., n_r)$ will at the end of the recursion be an optimal sequence.

This algorithm has in general complexity $O(n_r n_k^2)$. The factor n_k^2 is due to the maximization in step 2 that has to performed for each k. This part can be reduced to a factor n_k in our case. First find $l^*_{i-1} = \text{argmax}_l \delta_{i-1}(l)$ (which is an $O(n_k)$ operation). Because the second term $\beta I(k = l)$ only give contribution if $k = l$, we need for a given k only to compare $\delta_{i-1}(k) + \beta$ with $\delta_{i-1}(l^*_{i-1}) + \beta I(k = l^*_{i-1})$ which is an operation not depending on the number of classes. The total algorithm then has complexity reduced to $O(n_r n_k)$.

References

Ahuja, R. K., Magnanti, T. L. & Orlin, J. B. (1993), *Network flows: theory, algorithms, and applications*, Prentice-Hall, Englewood Cliffs, New Jersey.

Besag, J. (1974), 'Spatial interaction and the statistical analysis of lattice systems', *Journal of Royal Statistical Society, Series B* **36**, 192–236.

Besag, J. (1986), 'On the statistical analysis of dirty pictures', *Journal of Royal Statistical Society, Series B* **48**(3), 259–302.

Ferrari, P. A., Frigessi, A. & Gonzaga De SÁ, P. (1995), 'Fast approximative maximum a posteriori restoration of multicolour images', *Journal of Royal Statistical Society, Series B* **57**(3), 485–500.

Frankot, R. T. & Chellappa, R. (1987), 'Lognormal random-field models and their applications to radar image synthesis', *IEEE Trans. Geosc. Remote Sensing* **25**, 195–206.

Garey, M. & Johnson, D. (1979), *Computers and intractability. A guide to the theory of NP-completeness*, W.H. Freeman and company.

Geman, S. & Geman, D. (1984), 'Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6), 721–741.

Gray, A. J. (1993), 'Contribution: Meeting on MCMC methods', *Journal of Royal Statistical Society, Series B* **55**, 58–61.

Greig, D. M., Porteous, B. T. & Seheult, A. H. (1989), 'Exact maximum a posterior estimation for binary images', *Journal of Royal Statistical Society, Series B* **51**, 271–279.

Holmberg, K. (1996), personal communication.

Jörnsten, K. O. & Näsberg, M. (1986), 'A new Lagrangian relaxation approach to the generalized assignment problem', *European Journal of Operations Research* **27**, 313–323.

Nemhauser, G. L. & Wolsey, L. A. (1988), *Integer and Combinatorial Opimization*, Wiley.

Qian, W. & Titterington, D. M. (1989), 'On the use of Gibbs Markov chain models in the analysis of images based on second-order pairwise interactive distributions', *Journal of Applied Statistics* **16**(2), 267–281.

Solberg, A. H. S. & Jain, A. K. (1996), Feature selection applied to SAR imagery. Accepted for publication in IEEE Trans. of Geo. and Remote Sensing.

Swendsen, R. H. & Wang, J.-S. (1987), 'Nonuniversal critical dynamics in Monte Carlo simulations', *Physical Review Letters* **58**, 86–88.

Viterbi, A. J. (1967), 'Error bounds for convolutional codes and an asymptotically optimal decoding algorithm', *IEEE Trans. Informat. Theory* **IT-13**(3), 260–269.