EVOLUTIONARY GIBBS SAMPLER FOR IMAGE SEGMENTATION

Xiao Wang and Han Wang

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
hw@ntu.edu.sg

ABSTRACT

We propose a novel evolutionary algorithm for the function optimization problem in Bayesian image segmentation with Markov random field prior. Function variables are partitioned into several codings. A pivot coding is selected and variables in it are evolved respectively according to their probability distributions which encode both the evolutionary pressure and contextual constraints from neighboring pixels. Variables in other codings are evolved according to their conditional probabilities. In summary, the algorithm is about building probabilistic models to guide search. It achieves the efficiency and flexibility by incorporating Gibbs sampler in an evolutionary approach. Remarkable performance is observed in some experiments.

1. INTRODUCTION

We consider image segmentation as a labeling procedure through which a two dimensional set $L = \{l_{(i,j)} | \forall (i,j) \in S\}$ is to be estimated upon the image lattice $S = \{(i,j) | 1 \leq i,j \leq n\}$ and each $l_{(i,j)}$ takes its value from set $\{1, ..., M\}$ where M is the number of regions. Given the image data $D = \{d_{(i,j)} | \forall (i,j) \in S\}$, optimal segmentation is often defined by the *maximum a posteriori* (MAP) criterion:

$$L^* = argmax P(L|D) = argmax\{P(D|L) + P(L)\} \quad (1)$$

which induces function optimization in an intractable space with very high dimensionality (number of pixels) and complex landscape [1].

Simulated Annealing (SA) with Gibbs sampler [2] is widely used to attack the optimization problem. We assume L to be a Markov random field (MRF) such that conditional probability is easily calculated by

$$P(l_{(i,j)} | l_{N_{(i,j)}}) = \frac{e^{-U(l_{(i,j)} | l_{N_{(i,j)}})}}{\sum_{l_{(i,j)}} e^{-U(l_{(i,j)} | l_{N_{(i,j)}})}} \quad (2)$$

where $N_{(i,j)}$ is the set of neighboring pixels for (i,j) and $U(l_{(i,j)} | l_{N_{(i,j)}})$ is the local energy [1]. Starting from an initialization L^0, Gibbs sampler sequentially updates each variable $l^t_{(i,j)}$ into $l^{t+1}_{(i,j)}$ by sampling from the conditional posterior probability $P(l_{(i,j)} | D, l_{N_{(i,j)}})$. The cycle is repeated either a fixed number of iterations or until a quasi-equilibrium is reached. When combined with an annealing strategy, the above procedure is performed repeatedly, each time from the current L and from a lower temperature T. Transition of L to a "bad" state is less and less happened and eventually when T approaches 0, the system is "frozen" near the function maximum. Global convergence is theoretically guaranteed for SA with Gibbs sampler [2], but it also means significant computation.

In this paper we propose a hybrid algorithm which is an extension of our previous work on evolutionary optimization with MRF prior [3]. Just like a genetic algorithm [4], the algorithm also works on a population of potential solutions, called *chromosomes*. But it contains several novelties. There is neither crossover nor mutation involved. Function variables are evolved according to some probability distributions so that the algorithm appears more similar to an evolutionary search at the beginning while the very late stages will be characterized by Gibbs sampler-like local search. The paper is thus about an *Evolutionary Gibbs sampler* with application on image segmentation.

2. AN EVOLUTIONARY ALGORITHM

We partition the variable set L into several subsets L^k, named codings [5], such that no two variables in one coding are neighbors (see Fig. 1). There are two codings for the first-order and four codings for the second-order neighborhood system. A circle in the figure corresponds to a variable $l_{(i,j)}$ and variables marked by different numbers belong to different codings. From now on our study will be focused on the second-order neighborhood system (it can be extended to other neighborhood systems immediately). We consider the four codings as one pivot coding plus three other codings. The pivot coding may be specified arbitrarily; so suppose we pick coding 1. The coding 2, 3 and 4 are classified into other codings. The algorithm can be subsequently outlined by the following procedure.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

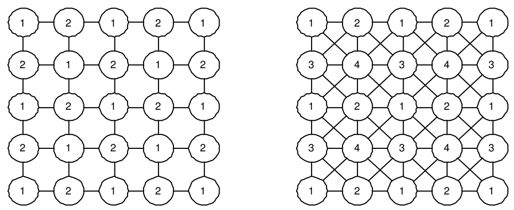

(a) (b)

Fig. 1. Codings. (a)first-order neighborhood. (b)second-order neighborhood.

1. set $t = 0$, randomly generate the initial population $popu(0)$.
2. set $t = t + 1$, $popu(t) = popu(t - 1)$
3. estimate a probability distribution of each variable in the pivot coding and update the variable by sampling from its distribution
4. update variables in other codings sequentially by sampling from their conditional probabilities.
5. check termination criteria. if satisfied, stop; otherwise go to step 2.

Each chromosome in the population will be an instance of the two-dimensional random field L (see Fig. 2). For an illustration, there are only two regions in the image and the pixels in the pivot coding are displayed in a grey background. For each of these chromosomes, the posterior energy $U(L|D)$ is a natural indication of its fitness. Since a lower energy means a higher probability and the energy value could be negative, we use a linear mapping $F(.) = U_{max} - U(.)$ in which $U(.)$ and $F(.)$ are the posterior energy and corresponding fitness evaluation of a chromosome while U_{max} denotes the maximum energy in the population.

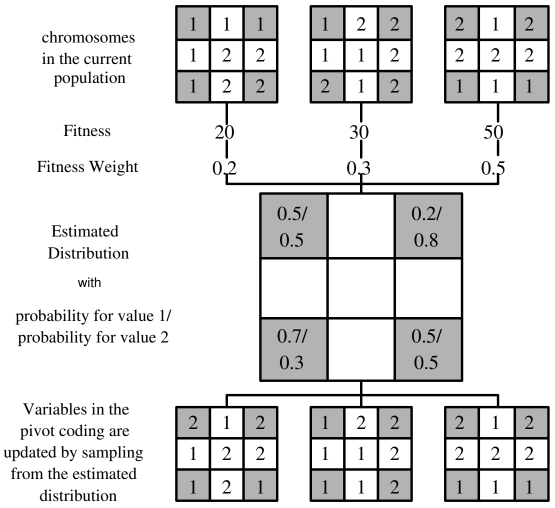


Fig. 2. Chromosomes and evolution of the pivot coding.

Now we let the variables in coding 2, 3 and 4 be fixed. Thus according to the Markovianity assumption, variables in coding 1 are probabilistically independent to each other. That means we can estimate their distributions separately. To do so, we compute a fitness-weight

$$FW(.) = \frac{F(.)}{\sum_{k \in popu(t)} F(k)} \tag{3}$$

for each chromosome in the population which has a fitness evaluation $F(.)$.

For a variable $l_{(i,j)}$ in the pivot coding, its probability to take value m in the next generation is then approximated by summing over the corresponding fitness-weight values for all chromosomes that currently have label m at pixel (i, j). That is,

$$P(l_{(i,j)}^{t+1} = m) = \sum_{k \in popu(t)} FW(k | l_{(i,j)}^{t} == m) \tag{4}$$

We use $P_{(i,j)}(m)$ to denote this probability. Then the vector $\overrightarrow{P}_{(i,j)} = [P_{(i,j)}(1), ..., P_{(i,j)}(M)]^T$ is an estimated distribution on the label set $\{1, ..., M\}$ and is computed for all the variables $l_{(i,j)}$. Each chromosome then has its variables in the pivot coding updated by new values sampled from their distributions $\overrightarrow{P}_{(i,j)}$ (see Fig. 2). Variables in coding 2, 3 and 4 are not displayed for clearance and actually they remain unchanged at this step.

Variables in other codings will be evolved in a sequential manner. First we let the variables in coding 1, 3 and 4 be fixed. Consequently for each chromosome in the population, the variables in coding 2 are probabilistically independent to each other and the conditional probabilities of them can be calculated separately by (2). According to each of these distributions, a new value is sampled and assigned to its associated variable. In this way we finish updating variables in coding 2 for this chromosome. The same operation is conducted on the other chromosomes to update the variables in coding 2 and after it all the variables in coding 2 in the population are updated. Variables in coding 3 and 4 will be updated similarly, e.g. we let variables in coding 1, 2 and 4 be fixed and update variables in coding 3 by their conditional probabilities. After the variables in coding 4 are also updated, the new chromosome appears as a whole.

The steps of updating pivot coding and other codings are repeated for several populations. The convergence is declared when the posterior energy of the best chromosome in the population has not changed over the past a few generations. Some preliminary experiments show that the convergence speed of the algorithm introduced so far is very slow. We have tried an *elitist* strategy, but it does not help very much. In the next section we propose to incorporate local search into the current approach and it is found to accelerate the convergence to a satisfactory extent.

3. HYBRID WITH LOCAL SEARCH

The necessity of incorporating local search is also advocated in a genetic algorithm for image segmentation [6]. The authors resort to the engineered conditioning (EC) operator which has two parameters: e_1 which determines how many chromosomes to be influenced and e_2 which denotes how many pixels in the chosen chromosome will be improved. e_1 and e_2 are given small value like 0.1 initially and are increased gradually towards 1 which means a large number of hill-climbing moves. The authors argue that in the later generations of evolution it is highly likely that most of the chromosomes in the population lie in the vicinity of an optimum and in this situation it is imperative for the algorithm to focus on local search.

The method proposed here is motivated by this idea, but it takes a more convenient way. The key to our method is the probability distribution (4) used for updating the pivot coding. Currently it is estimated from the fitness values and encodes only the selective pressure for evolution. We use $\overrightarrow{P}^{evo}_{(i,j)}$ to denote it.

Now suppose a totally different scheme is adopted where each chromosome has its own distributions for the pivot coding which only encode the contextual constraints; that is, each variable in coding 1 are updated by sampling from its conditional probability. We use $\overrightarrow{P}^{con}_{k;(i,j)}$ to denote it which is calculated by (2) for chromosome k. The algorithm then turns to be a multi-start Gibbs sampler [2], i.e. each chromosome is subject to a series of local improvements based on its own statistics. A simple yet effective way to incorporate such Gibbs sampler-like local search is to use a weighted sum of these two distributions:

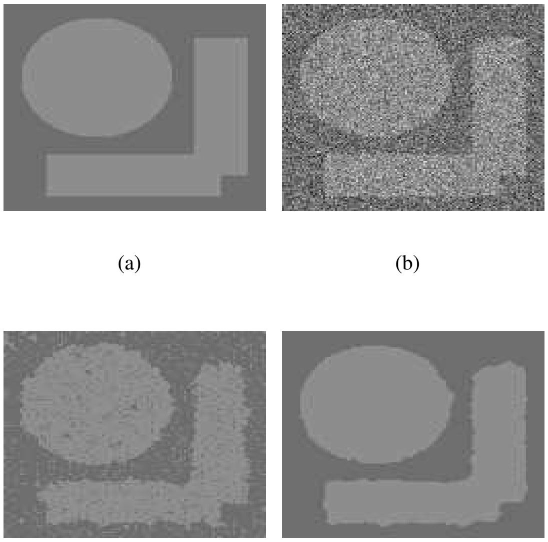
$$\overrightarrow{P}_{k;(i,j)} = \lambda \times \overrightarrow{P}^{con}_{k;(i,j)} + (1 - \lambda) \times \overrightarrow{P}^{evo}_{(i,j)} \qquad (5)$$

in which $\lambda \in [0, 1]$. Inspired by the EC operator [6], λ could be adjusted to embody different judgements - which distribution should be used more, the evolutionary one or the context-conditional one. If λ is given a small value, $\overrightarrow{P}^{evo}_{(i,j)}$ will possess a big weight and the evolutionary global search is paid more attention; otherwise when λ is set high, $\overrightarrow{P}^{con}_{k;(i,j)}$ is more weighted and the context based local search is focused.

How to tune the parameter λ is a problem that comes along. A lot of heuristics of parameter control exist in the literature (see survey [7]). In our implementation a linear model is employed:

$$\lambda(i) = i/\text{NUM\_GENS}, \ \text{for } i = 0 \ldots \text{NUM\_GENS} \qquad (6)$$

where NUM_GENS denotes the number of generations predefined. Initially $i = 0$ and the algorithm is totally evolutionary. At the last generation where $i =$ NUM_GENS, all chromosomes are subject to local search only. Our algorithm is hence considered as a hybrid of those two kinds of search.



(a) (b)

(c) (d)

Fig. 3. Hybrid with local search. (a)original image. (b)noisy image. (c)segmentation by pure evolutionary search. (d)segmentation by hybrid search.

An experimental comparison is included here to demonstrate the effect of local search on helping convergence. The task is to segment a degraded image (Fig. 3(b)). The results by performing a pure evolutionary search and a hybrid search both over 50 generations are shown in Fig. 3(c,d). A visual comparison of reconstruction quality shows that a hybrid algorithm does much better. For a further study, Fig. 4 depicts the corresponding curves of convergence (posterior energy/generation). While a pure evolutionary search appears to demand a large number of generations of computation, a hybrid search is seen to have a desirable convergence rate.

4. EXPERIMENTAL RESULTS

In this section we present some experiments on noisy or textured image segmentation which is extensively studied

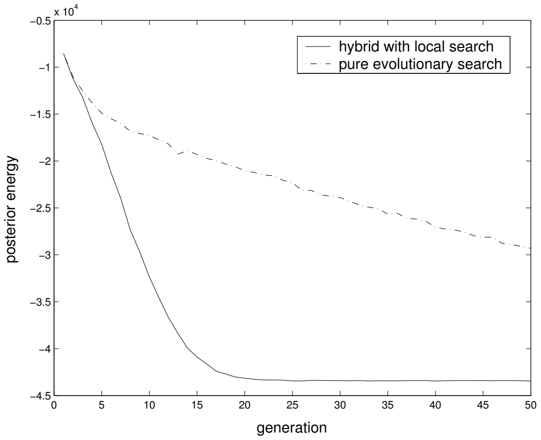


Fig. 4. Convergence curves with and without local search.

in the MAP-MRF framework. The hierarchical generalized Ising model [8] is used and we assume the model parameters are known beforehand. The population size is set to be 60. There are no other parameters like the crossover rate or mutation rate as in a GA.

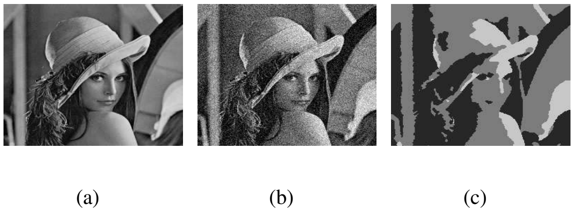


Fig. 5. Segmentation of noisy image. (a)original image. (b)noisy image. (c)segmentation result.

In Fig. 5, original image is corrupted by i.i.d Gaussian noise with standard deviation 25 and we perform a three-level segmentation of gray value 40, 125 and 200. The other experiment is on a textured image with handwritten numbers (Fig. 6). The perfect segmentation is depicted to be a comparison. In both experiments we observe quite good segmentations in just a few generations of evolution.

5. CONCLUSION

Evolutionary optimization with Markov random field prior is studied in this work. We develop a novel algorithm in which the population evolution is achieved not by genetic recombination but by sampling from some probabilistic distributions. While variables in other codings are subject to their conditional probabilities, variables in the pivot codings are subject to a combined pressure of natural selection and contextual constraints. The algorithm is then seen to combine together the power of evolutionary search on global exploration and the efficiency of Gibbs sampler on local exploitation. It is testified to be useful in MRF model-based image segmentation.

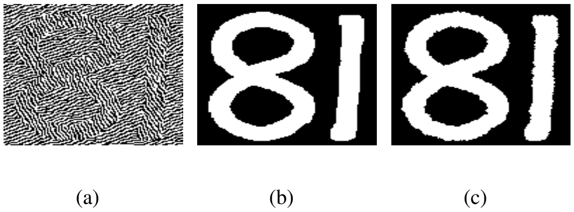


Fig. 6. Segmentation of texture image. (a)original image. (b)perfect segmentation. (c)segmentation result.

6. REFERENCES

[1] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, Springer-Verlag, 2001.

[2] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[3] X. Wang and H. Wang, “Evolutionary optimization in markov random field modeling,” in *Proc. 4th International Conference on Information, Communications and Signal Processing*, 2003.

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[5] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society, Series B*, vol. 36, pp. 192–236, 1974.

[6] S.M. Bhandarkar and H. Zhang, “Image segmentation using evolutionary computation,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 1–21, Apr. 1999.

[7] A.E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul. 1999.

[8] H. Derin and H. Elliott, “Modeling and segmentation of noisy and textured images using Gibbs random fields,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 1, pp. 39–55, Jan. 1987.