Summary

Markov random field(MRF) modelling is a popular pattern analysis method. It can be used in texture classification and segmentation as well as image restoration. MRF parameter estimation plays an important role in MRF modelling. In order to estimate MRF parameter effectively and efficiently, an MRF parameter estimation method based on MCMC method is proposed in this paper. A Markov chain is constructed to sample the MRF parameters via Monte Carlo method. MLL model is used as image model. In order to avoid to calculate the normalizing partition function, pseudo-likelihood function is used to represent likelihood function. Compared to least squares fit method, our method is more accurate and can be used for multi-graylevel texture parameter estimation effectively according to the experiments in the paper. This method can be extended to be used in multiresolution analysis of texture modeling and segmentation of textured images.

MRF Parameter Estimation by MCMC Method

Lei Wang   Jun Liu   Stan Z.Li

School of Electrical and Electronic Engineering

Nanyang Technological University

Singapore 639798

p143572659@ntu.edu.sg, ejliu@ntu.edu.sg szli@szli.eee.ntu.edu.sg

Draft submitted: January 13, 1999

First Revised: July 18, 1999

Abstract

Markov random fields (MRFs) have been used in image restoration, texture classification and segmentation as a prior model of pattern distributions. An important issue in MRF modeling is MRF parameter estimation. In this paper, an MRF parameter estimation method is proposed for effective and efficient estimation of MRF parameters. The method is based on Markov Chain Monte Carlo (MCMC) in which a Markov chain is constructed to sample the MRF parameters using Monte Carlo method. Experiments show that the proposed method is more accurate than the least squares fit method, and can be used for effective estimation of multi-graylevel texture parameters. This method can be extended for the use in multiresolution analysis of texture modeling and segmentation of textured images.

Key Words: MRF, MCMC, Least Squares Fit, Parameter Estimation, Pseudo-likelihood

1 Introduction

Mathematical modeling in pattern analysis is aimed to extract the intrinsic characteristics of the pattern in a few parameters so as to represent the pattern effectively. Markov random field modeling has been a useful tool in pattern recognition and computer vision [1, 2, 3]. The most important characteristic of MRF modeling is that the global patterns can be formed via stochastic propagation of local interactions.

MRF parameters have to be given or estimated for a complete specification of an MRF model. During the past years, many authors presented methods to estimate MRF parameters, including simulated annealing[4], maximum likelihood[5], coding method[6], mean field approximations[7], Bayesian estimation[8] and least squares (LS) fit[9], and a more recent work using reversible jumps [10]. An overview can be found in [3].

Least squares(LS) methods and maximum likelihood methods are often used. However, LS is not accurate in estimation and maximum likelihood method is time-consuming. Here a method based on MCMC is used to estimate the parameters which can give a good solution to the estimation.

The general parameter estimation principle is as follows. Let F denote any finite set which comprises of a random field and $f \in F$ is a observation of F. On F a family of distributions

$$\Pi = \{\Pi(F;\theta) : \theta \in \Theta\}$$

is considered where $\Theta \subset R^d$ is a set of parameters. The 'true' parameter $\theta^* \in \Theta$ is not known and needs to be determined or at least approximated. The only available information is hidden in the observation f which is a realization of F. Now the problem is how to choose $\hat{\theta}$ as a substitute for θ^* if f is picked at random from $\Pi(F;\theta^*)$.

In this paper, the estimation of parameters is based on deriving posterior distribution calculated using the Metropolis-Hastings algorithm. This is a Markov chain

Monte Carlo(MCMC) technique[11].

The paper is arranged as follows. MRF image model is discussed in section 2. MCMC parameter estimation is proposed in section 3. The experiments are shown in section 4 and conclusion is given in section 5.

2 MRF Modeling

This section introduces some notations related to MRF modeling which will be used in the following sections of the paper.

A lattice is a square array of pixels, or sites, $\{(j,k) : 0 \leq j \leq N-1, 0 \leq k \leq N-1\}$. We adopt a simple numbering of sites by assigning sequence number $i = k + Nj$ to site (j,k). Letting $M = N^2$ denote the number of sites,

$$S = \{0, 1, ..., M-1\}$$

index the set of sites. A random field model is a distribution for the M-tuple random vector F, which contains a random variable $F(i)$ for the value of site i. The sites in S are related to one another via a neighborhood system. A neighborhood system for S is defined as

$$\mathcal{N} = \{\mathcal{N}_i | \forall i \in S\}$$

where $\mathcal{N}_i$ is the set of sites neighboring i. The neighboring relationship has the following properties:

(1) a site is not neighboring to itself;

(2) the neighboring relationship is mututal.

A clique c is a set of sites in which all pairs of sites are mutual neighbors. The set of all cliques in a neighborhood system is denoted as Q.

Suppose F is an MRF. Let $f \in F$ be a realization of F. A clique function, or potential function, $V_c(f)$, is associated with each clique and the energy function, $U(f)$, of MRF can be expressed as the sum of clique functions.

$$U(f) = \sum_{c \in Q} V_c(f)$$

To a homogeneous MRF, the potential function is independent of locations. Thus the number of clique potentials can be reduced to the number of clique types, that is, each potential corresponding to a clique type.

Consider a multi-level logistic (MLL) model[3]. Let $\mathcal{L} = \{1, ..., m\}$ be the label set and $\theta = (\theta_1, ..., \theta_n)$ be the parameter vector for clique potentials where each component corresponds to a clique type. Consider the distribution of Gibbs form,

$$P(F = f, \theta) \propto P(F = f|\theta) = Z(\theta)^{-1} \exp(-U(f, \theta)) \tag{1}$$

where $U(f, \theta)$ is energy function and depends linearly on θ. Suppose $H(f) = (H_1, ..., H_n)$ is the histogram of cliques of f, n denotes the index of clique type. Let

$$\delta(z) = \begin{cases} 1, & \text{if } z = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$H_i = \sum_{j \in S} (2 \prod_{j' \in c_i(j)} \delta(f_j - f_{j'}) - 1), i = 1, ..., n$$

where $c_i(j)$ denotes the neighbor of site j in the i-th clique c_i.

Then the distributions have the form

$$P(f,\theta) = Z(\theta)^{-1}\exp(<\theta, H(f)>),\ \theta \in \Theta$$

where $<\theta, H(f)>=\sum_{i=1}^{n}\theta_i H_i$ is the inner product of θ and H, and $Z(\theta) = \sum_f \exp(<\theta, H(f)>)$ is the normalizing partition function. The conditional probability is as follows

$$P(f_j|f_{N_j}) = \frac{\exp(-<\theta, H(f_j)>)}{\sum_{z_j\in\mathcal{L}}\exp(-\sum<\theta, H(z_j)>)} \tag{2}$$

where $H(f_j)$ is the local histogram only calculated in the neighborhood of site j. $H(z_j)$ denotes the local histogram replacing f_j with z_j and the neighborhood of j is fixed. The computation of $Z(\theta)$ is infeasible because there are a combinatorial number of elements in the configuration space. In order to avoid using the partition function $Z(\theta)$, the pseudo-likelihood function

$$\begin{aligned} PL(f|\theta) &= \log(\prod_{j\in S} P(f_j|f_{N_j})) \\ &= \sum_{j\in S}(<\theta, H(f_j)> - \log\sum_{z_j\in\mathcal{L}}\exp(<\theta, H(z_j)>)) \end{aligned} \tag{3}$$

can be used to replace likelihood function. The pseudo-likelihood does not involve the partition function $Z(\theta)$. Hence it is much easier to be calculated.

3 MCMC estimation of MRF parameters

According to Bayesian theorem, the posterior distribution of θ conditional on f is

$$P(\theta|f) = \frac{P(\theta)P(f|\theta)}{\int P(\theta)P(f|\theta)\mathrm{d}\theta} \propto P(\theta)P(f|\theta) \tag{4}$$

According to Gilks, Richard and Spiegelhalter[11], any features of the posterior distribution are legitimate for Bayesian inference: moments, quantiles, highest posterior density regions, etc. All these quantiles can be expressed in terms of posterior expectations of functions of θ. The posterior expectation of a function $g(\theta)$ is

$$E[g(\theta)|f] = \frac{\int g(\theta)P(\theta)P(f|\theta)\mathrm{d}\theta}{\int P(\theta)P(f|\theta)\mathrm{d}\theta} \tag{5}$$

The integrations in this expression are difficult to be solved in Bayesian inference. Monte Carlo integration including Markov Chain Monte Carlo (MCMC) approach[8] can be used to deal with the difficulty[11]. The task is to evaluate the expectation

$$E(g(\theta)) = \frac{\int g(\theta)P(\theta)\mathrm{d}\theta}{\int P(\theta)\mathrm{d}\theta} \tag{6}$$

A Markov chain can be adopted for the purpose of evaluation. Suppose we generate a sequence of random variables $\{\theta^0, \theta^1, ...\}$. At each time $t \geq 0$, the next state θ^{t+1} is sampled from a distribution $P(\theta^{t+1}|\theta^t)$ which depends only on the current state θ^t of the chain. This Markov chain is assumed to be time-homogeneous. Thus the sequence will gradually converge to a unique stationary distribution $\phi(.)$. After a sufficient long burn-in of say m iterations, $\{\theta^t, t = m+1, ..., n\}$ will be dependent samples approximately from $\phi(.)$. Let

$$\overline{\theta} = \frac{1}{n-m} \sum_{t=m+1}^{n} \theta^t \tag{7}$$

This is an ergodic average. Convergence to the required expectation is ensured by the ergodic theorem. Eq.(7) shows how a Markov chain can be used to estimate $E(\theta|f)$. Such a Markov chain can be constructed by Metropolis-Hastings algorithm[11]. At each time t, the next state θ^{t+1} is chosen by first sampling a candidate point θ' from a proposal distribution $q(\theta'|\theta^t)$.

The choice of proposal distribution is almost arbitrary; here a multivariate normal distribution centered on the current value θ^t is adopted. The candidate θ' is accepted with probability

$$\alpha(\theta^t, \theta') = \min(1, \frac{P(\theta'|f)q(\theta'|\theta^t)}{P(\theta^t|f)q(\theta^t|\theta')}) \tag{8}$$

If the candidate θ' is accepted, the next state becomes $\theta^{t+1} = \theta'$, otherwise $\theta^{t+1} = \theta^t$. Since $P(\theta|f) \propto P(\theta)P(f|\theta)$ and the prior $P(\theta)$ can be assumed to be flat when the prior information is totally unavailable,

$$\begin{aligned} \alpha(\theta^t, \theta') &= \min(1, \frac{P(\theta'|f)q(\theta'|\theta^t)}{P(\theta^t|f)q(\theta^t|\theta')}) \\ &= \min(1, \frac{P(f|\theta')q(\theta'|\theta^t)}{P(f|\theta^t)q(\theta^t|\theta')}) \end{aligned} \tag{9}$$

Since the choice of proposal distribution here is normal centered on the current value, $q(\theta'|\theta^t) = q(\theta^t|\theta')$ due to the symmetric property of the proposed distribution. Thus the acceptance probability formula can be reduced to

$$\alpha(\theta^t, \theta') = \min(1, \frac{P(f|\theta')}{P(f|\theta^t)}) \tag{10}$$

Thus the Metropolis-Hastings algorithm is switched to Metropolis algorithm. When we use pseudo-likelihood to represent the likelihood function, we get

$$\begin{aligned} \alpha(\theta^t, \theta') &= \min(1, \exp(PL(f|\theta') - PL(f|\theta^t))) \\ &= \min(1, \exp(\sum_{j \in S}((< \theta', H(f_j) > - < \theta^t, H(f_j) >) \\ &\quad - \log(\sum_{z_j \in \mathcal{L}} \exp(< \theta', H(z_j) >)) - \log(\sum_{z_j \in \mathcal{L}} \exp(< \theta^t, H(z_j) >))) \end{aligned} \tag{11}$$

With this acceptance probability, the θ can be approximated effectively.

The Metropolis-Hastings algorithm can be summarized in the following procedures:

Initialize θ^0; set $t = 0$ and $T =$ maximum number of iteration

While $t < T$

BEGIN

Sample a point θ' from $q(.|\theta^t)$

Sample a uniform $(0, 1)$ random variable v

If $v \leq \alpha(\theta^t, \theta')$, set $\theta^{t+1} = \theta'$. Otherwise set $\theta^{t+1} = \theta^t$

Increment t

END

The transition kernel for the Metropolis-Hastings algorithm is

$$P(\theta^{t+1}|\theta^t) = q(\theta^{t+1}|\theta^t)\alpha(\theta^t, \theta^{t+1}) + I(\theta^{t+1} = \theta^t)[1 - \int q(\theta'|\theta^t)\alpha(\theta^t, \theta')d\theta'] \qquad (12)$$

where $I(.)$ denotes the indicator function(taking 1 when its argument is true, and 0 otherwise). The first term in Eq.(12) arises from acceptance of a candidate and the second term arises from rejection for all candidates. Using the fact that

$$\pi(\theta^t)q(\theta^{t+1})|\theta^t)\alpha(\theta^t, \theta^{t+1}) = \pi(\theta^{t+1})q(\theta^t|\theta^{t+1})\alpha(\theta^{t+1}, \theta^t)$$

which follows from Eq.(8), the detailed balance equation can be given as

$$\pi(\theta^t)P(\theta^{t+1}|\theta^t) = \pi(\theta^t|\theta^{t+1})$$

This implies that if θ^t is from $\pi(.)$, then θ^{t+1} will be also. Thus, once a sample from the stationary distribution has been obtained, all subsequent samples will be from that distribution.

4 Experiments

In order to inspect the performance of the method proposed in this paper, a Gibbs sampler[3] is used to sample textures with the specified parameters. Here a second-order neighborhood system is used and four double-site cliques $\theta = \{\beta_1, \beta_2, \beta_3, \beta_4\}$ corresponding to $0°, 90°, 45°, 135°$ individually are

adopted as non-zero parameters. Fig.1 shows four 128×128 textures generated from the Gibbs Sampler. The first two textures are sampled with two graylevels and the next two textures are sampled with four graylevels. The parameters of the four textures are listed in Table 1. In order to get acceptable parameters, the MCMC procedure described in the previous section should be repeated until the Markov chain is converged. In our experiments, θ^0 are chosen randomly. The chain $\{\theta^0, \theta^1, ..., \theta^t\}$ will gradually "forget" its initial state θ^0 subject to regularity conditions and the chain will eventually converge to a unique stationary distribution which does not depend on t or θ^0 according to the discussion in the previous section.

Three independent samples of Markov chains for texture 4 are given in Fig.2 to Fig.5 where the horizontal axe is the iteration number of the Monte Carlo algorithm, and the vertical axe is the parameters $\{\theta_i, i = 1, ...4\}$ respectively. From the figures, we observe that the length of burn-in depends on θ^0. An informal approach to detection of convergence is visual inspection of plots of the Monte-Carlo output $\{\theta^t, t = 1, ..., n\}$. The Markov chains converge in less than 300 iterations in most examples according to visual inspection of the monitoring statistics. Here we set burn-in $m = 500$. More formal methods for convergence diagnostics can be found in[12][13].

Decision about the iteration number is an important and practical matter. The aim is to run the chain long enough to obtain adequate precision in the estimator. Here three chains are run in parallel with different starting values $\overline{\theta}$ from Eq.(7). If they do not agree adequately, the iteration number n must be increased. Initially, we set $n = 1000$. If the estimates $\overline{\theta}$ do not agree adequately, we increase 500 iterations each time until estimates are similar. We only need to inspect the mean $\overline{\theta}$ and variance σ of the Monte-Carlo output. In our experiments in Table 1, $n = 1000$ is enough. The results of MCMC approach in Table 1 are acceptable where σ denotes the average standard deviation of Markov chains after burn-in.

In order to verify the performance in accuracy, least squares(LS) fit method proposed by Derin and Elliott[9] is also used in our experiments. From Table 1, it can be seen that LS method is effective only to the textures with two graylevels, while MCMC method is effective to all examples in the experiments even more graylevels are adopted in the model. From the comparison, MCMC method proposed in this paper is much better than LS method. The MCMC routines are run on a Sun Ultra 2 workstation, each analysis takes less than 3 minutes to perform 1000 iterations.

5 Conclusion

A new method is proposed for MRF parameter estimation. In this method, a Markov chain is constructed to sample the MRF parameters via Monte Carlo method. Experiments with multi-graylevel texture images show that the proposed method produces more accurate estimates than the least squares fit method. MLL is used as the image model and pseudo-likelihood function is used to represent likelihood function. An investigation is being done to extend the method for multiresolution analysis of texture modeling and segmentation of textured images.

6 Acknowledgements

We wish to thank the constructive comments and suggestions of the reviewers.

References

[1] K. V. Mardia, ed., *Journal of Applied Statistics*, vol. 16(2). Special Issue on Statistic Image Analysis, 1989. An extended edition appears in K.V. Mardia and G.K. Kanji (ed) (1993), *Statistics and Images: 1*, Carfax.

[2] R. Chellappa and A. Jain, eds., *Markov Random Fields: Theory and Applications.* Academic Press, 1993.

[3] S. Li, *Markov random field modeling in computer vision.* New York: Springer-Verlag, 1995.

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoratopn of images," *IEEE Trans. PAMI*, vol. 6, no. 6, pp. 721–741, 1984.

[5] Y. Ogata and M. Tanemura, "Estimation for interaction potentials of spatial point patterns through the maximum likelihood procedure," *Annals of the Institute of Statistical Mathematics, B*, vol. 33, pp. 315–338, 1981.

[6] 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.

[7] D. Chandler, *Introduction to Modern Statistical Mechanics.* Oxford University Press, 1987.

[8] R. Aykroyd, "Bayesian estimation for homogeneous and inhomogeneous Gaussian random fields," *IEEE Trans. PAMI*, vol. 20, no. 5, pp. 533–539, 1998.

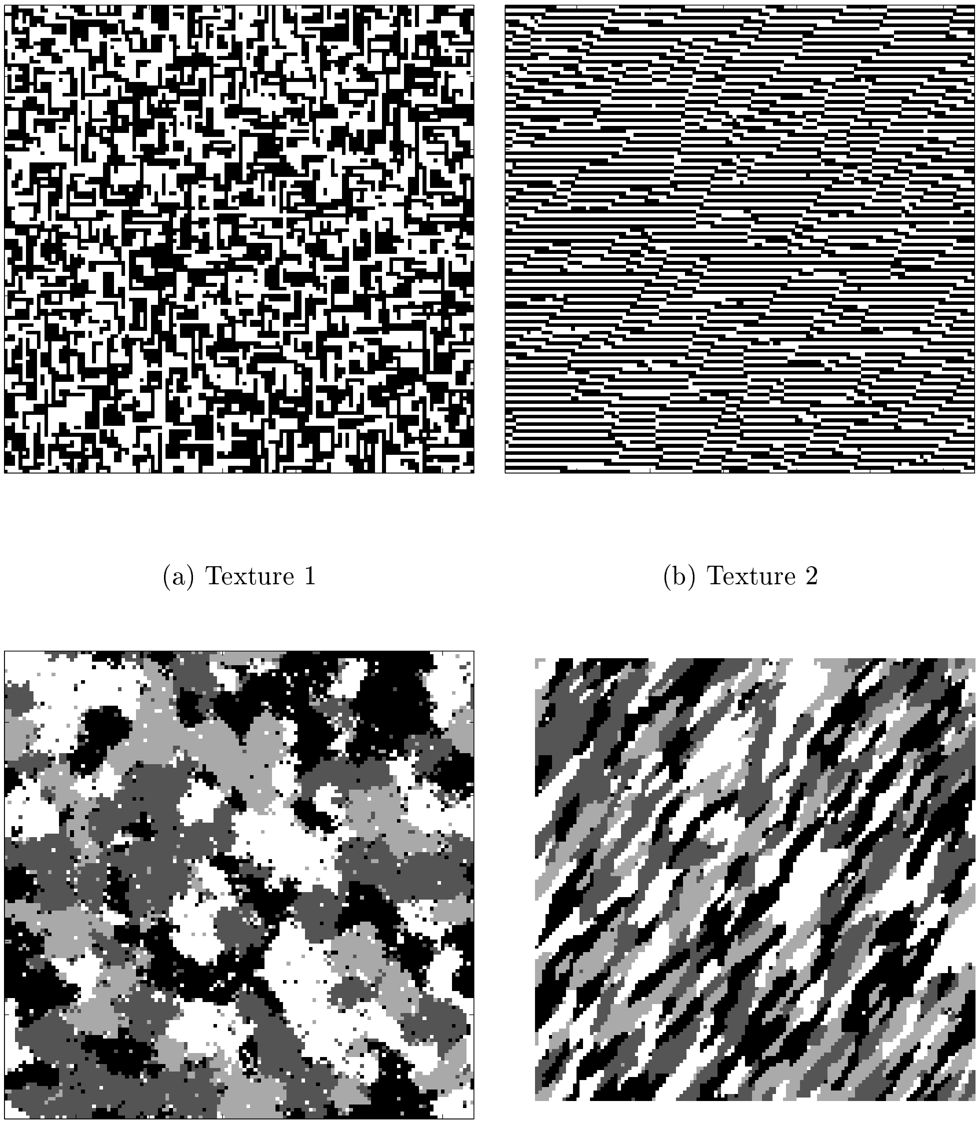
[9] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using gibbs random fields," *IEEE Trans. PAMI*, vol. 9, no. 1, pp. 39–55, 1987.

[10] S. Barker, *Image segmentation using Markov random field models.* PhD thesis, University of Cambridge, 1998.

[11] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in Practice*. Chapman& Hall, 1996.

[12] M. Cowles and B. Carlin, “Markov Chain Monte Carlo convergence diagnostics: a comparative review,” tech. rep., Division of Biostatistics, School of public health, University of Minnesota, 1994.

[13] S. Brooks, P. Dellaportas, and G. Roberts, “An approach to diagnosing total variation convergence of MCMC algorithms,” *University of Cambridge, http://www.stats.bris.ac.uk/ maspb/mypapers/brodr96.html*, 1996.



(a) Texture 1

(b) Texture 2

(c) Texture 3

(d) Texture 4

Figure 1: Textures used in the experiment. (a) Number of graylevels $M = 2$, (b) $M = 2$, (c) $M = 4$, (d) $M = 4$

Table 1: MRF parameter estimation

Textures	Method	β_1	β_2	β_3	β_4
Texture 1	specified	1	1	-0.5	-0.5
	LS	0.8448	0.8734	-0.4332	-0.4382
	MCMC	0.9884	0.9899	-0.5076	-0.5078
	σ	0.0436	0.0323	0.0278	0.0400
Texture 2	specified	1	-0.8	0.5	-0.5
	LS	0.9949	-0.8157	0.4960	-0.3244
	MCMC	1.0093	-0.8586	0.5569	-0.4522
	σ	0.0147	0.0235	0.0168	0.0245
Texture 3	specified	0.3	0.3	0.3	0.3
	LS	0.1152	0.1520	0.1867	0.1444
	MCMC	0.3478	0.2762	0.2960	0.2877
	σ	0.0266	0.0165	0.0130	0.0086
Texture 4	specified	0.5	1	-0.5	0.7
	LS	0.0415	0.4364	0	0.4201
	MCMC	0.5525	1.0394	-0.5951	0.6810
	σ	0.0321	0.0364	0.0490	0.0156

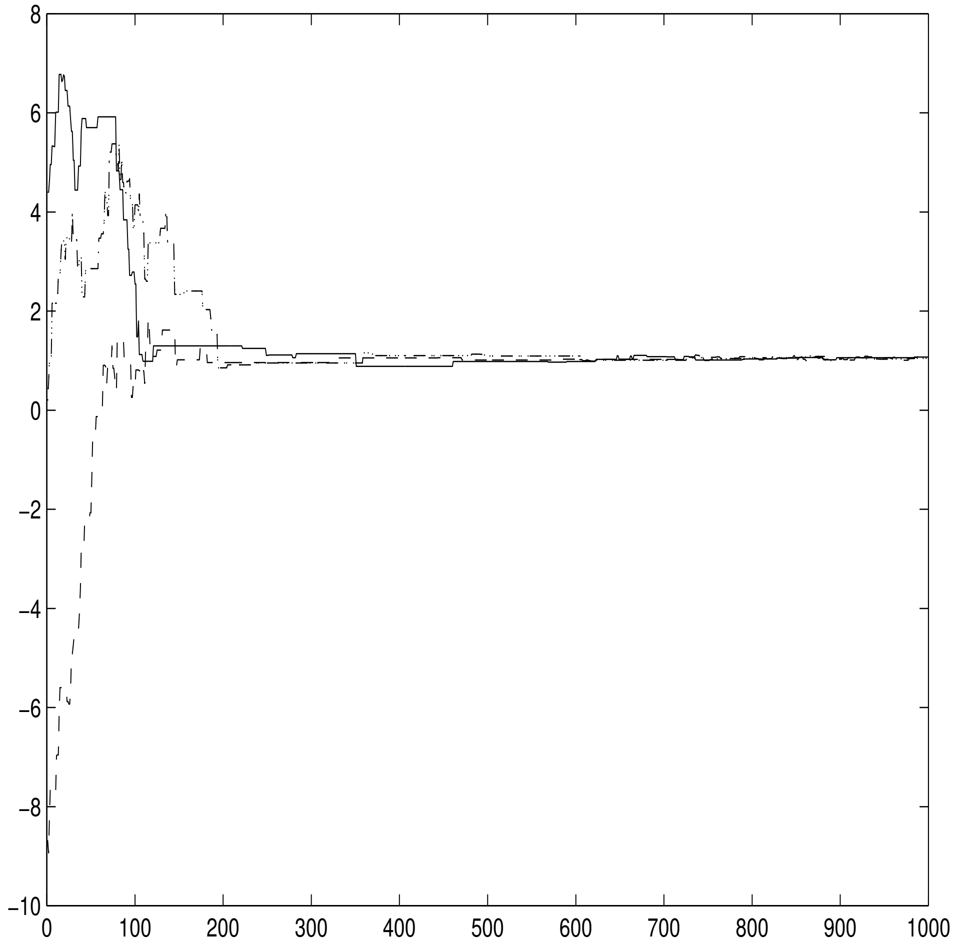


Figure 2: 1000 iterations with different starting values for estimating β_1 for texture 4

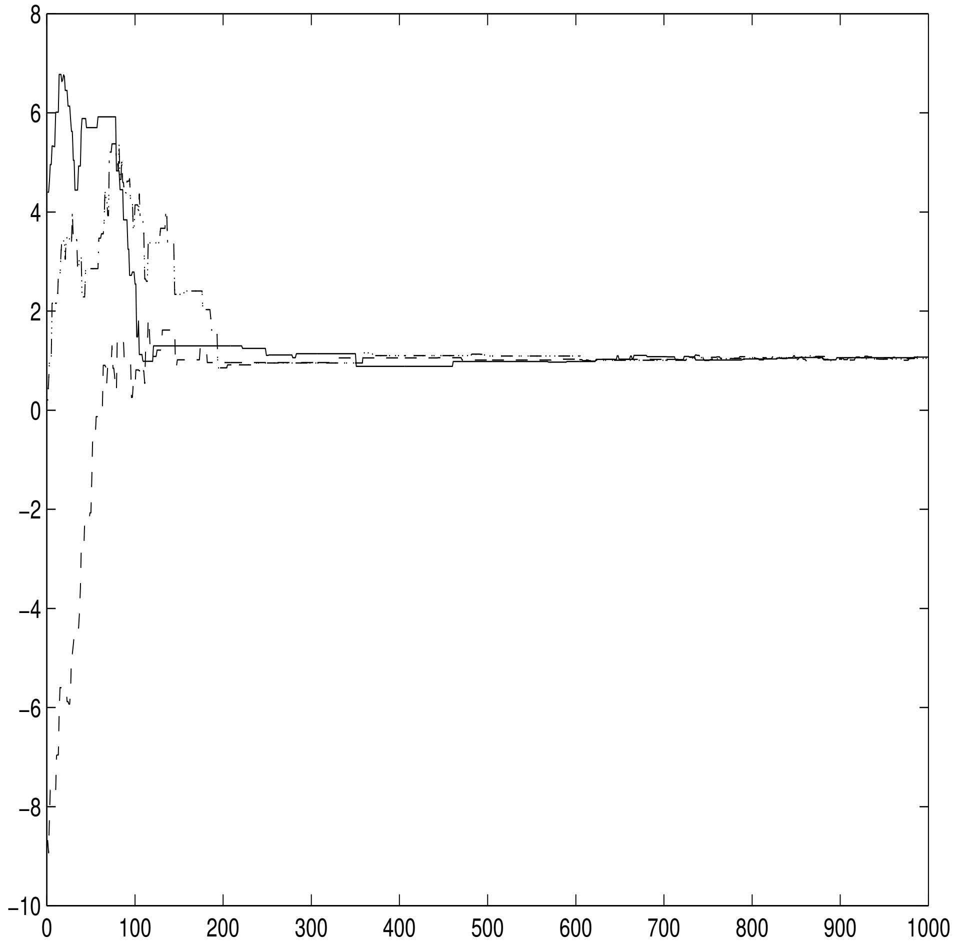


Figure 3: 1000 iterations with different starting values for estimating β_2 for texture 4

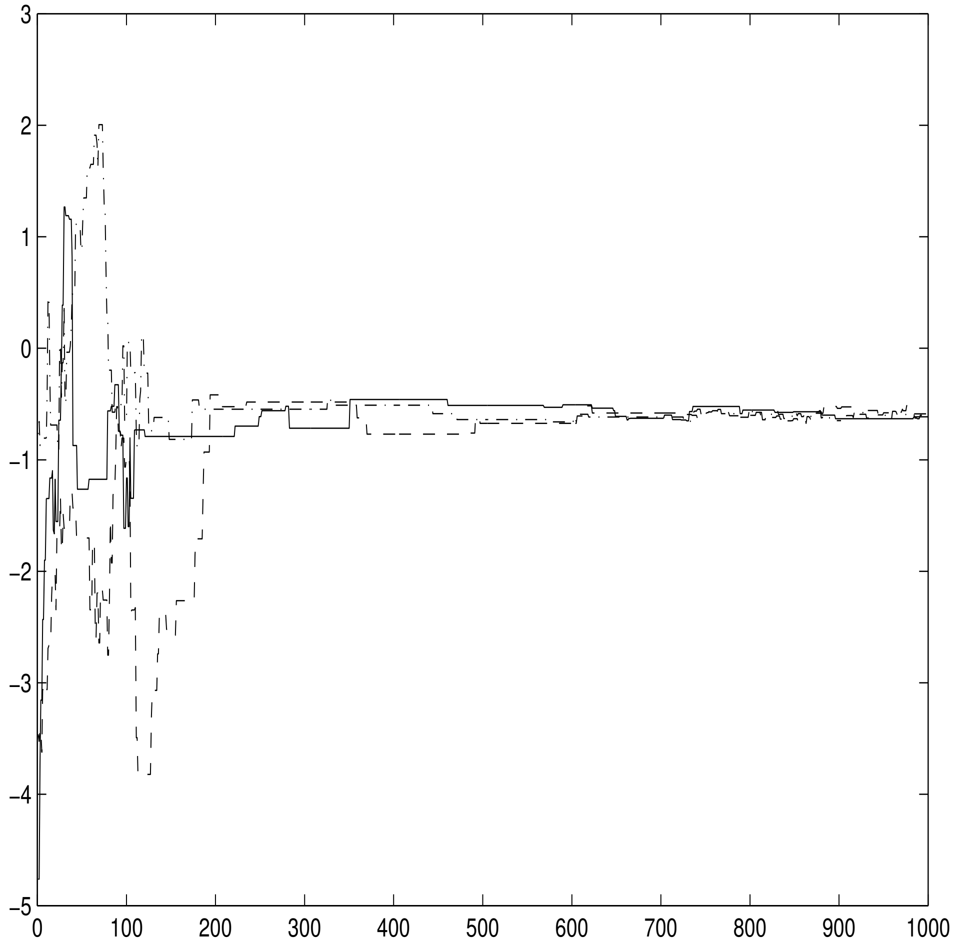


Figure 4: 1000 iterations with different starting values for estimating β_3 for texture 4

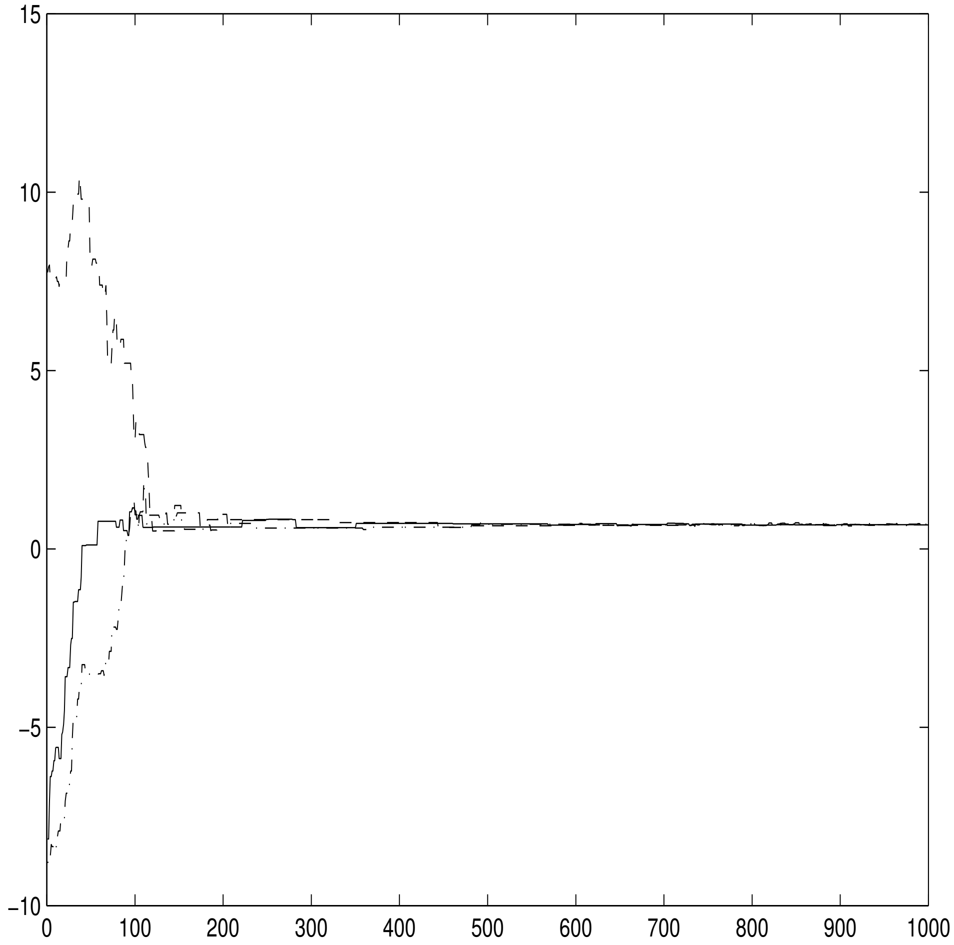


Figure 5: 1000 iterations with different starting values for estimating β_4 for texture 4