
COMPUTER EXERCISE 7
DISCRETE MARKOV RANDOM FIELDS
STATISTICAL IMAGE ANALYSIS, TMS016

1 Introduction

The purpose of this computer exercise is to give provide some intuition for how discrete Markov random field models behave. This will be done by simulating models with different parameters and see how the parameters affect the simulations. We will use these models for classification of images in the next computer exercise.

Before you begin, download the Matlab files for the exercise from the course homepage. When in doubt about how to use a specific function in Matlab, use `help` and `doc` to get more information.

Throughout the exercise, we will look at MRFs \mathbf{x} on an $m \times n$ lattice, taking values in $\{1, \dots, K\}$. The distribution of such a model can be written as

$$\pi(\mathbf{x}) = \frac{1}{Z} \exp \left(\sum_{i=1}^{mn} \left(\alpha_{x_i} + \frac{1}{2} \sum_{j \in N_i} \beta_{x_i, x_j} \right) \right).$$

Here $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)^T$ is a vector with parameters that determine the intensities for each class, and

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_{11} & \cdots & \beta_{1K} \\ \vdots & \ddots & \vdots \\ \beta_{K1} & \cdots & \beta_{KK} \end{pmatrix}$$

is a matrix with parameters that determine the strength of the spatial interactions in the model. The function `mrf_sim` can be used to simulate from this model using Gibbs sampling. In the function, the neighborhoods N_i are described by a stencil N . For example

$$N = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

says that each pixels has its four closest neighbors in its neighborhood N_i .

2 The Ising model

The most famous special case of the general model above is the Ising model. The model is named after Ernst Ising, who was a German physicist and developed the model as a model for ferromagnetism in statistical mechanics. It is obtained by choosing $K = 2$, $\boldsymbol{\alpha} = \mathbf{0}$,

$$\boldsymbol{\beta} = \begin{pmatrix} \beta & 0 \\ 0 & \beta \end{pmatrix}.$$

and the neighborhood matrix shown above.

- Study how the samples behave when changing the value of β . In the model formulation one can see that positive values will favor simulations where the pixels have value equal to the dominant class among the neighbors. A reason for why this model is so popular for theoretical studies is that it has a phase-transition at a value β^* where for values $\beta > \beta^*$, the distribution is highly bimodal also as the size of the lattice tends to infinity. Can you determine what the value of β^* is by looking at the simulations (the exact value of β^* is known)?

- How many iterations of the Gibbs sampler are needed before the simulations converge to something that seems to have a stable distribution? Measuring convergence for Gibbs samplers can be done in many ways. One way is to look at the convergence of the log-likelihood of the samples. One of the outputs from `mrf_sim` is the trajectory of the unnormalized (ignoring the constant Z) log-likelihood of the simulations.

3 More general binary MRFs

- Study how the sample behave if using $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ instead of $\boldsymbol{\alpha} = \mathbf{0}$. Note that if $\beta = 0$, then the probability for the first class at each pixel is given by

$$\frac{\exp(\alpha_1)}{\exp(\alpha_1) + \exp(\alpha_2)}.$$

- Study how the samples behave when changing the neighborhood structure N . You can for example test some of the following matrices

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

4 The Potts model

The generalization of the Ising model to $K > 2$ is usually called the Potts model.

- Try simulating a MRF with some $K > 2$ and study how changing β affects the simulations.
- Try using some more general dependence structure. Such as a diagonal matrix

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 & & \\ & \ddots & \\ & & \beta_K \end{pmatrix}$$

or a completely general $\boldsymbol{\beta}$ matrix.