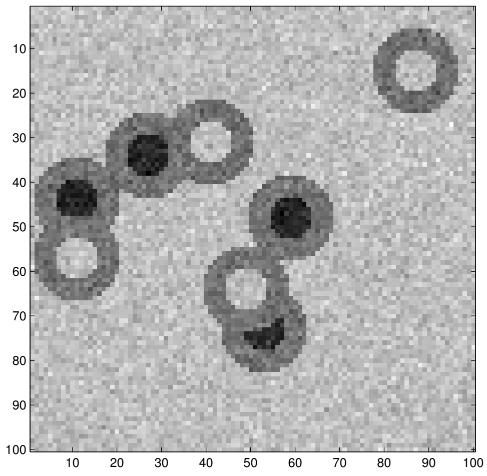
BUC5 : Computing 3

The following picture shows a (synthetic) noisy scene containing 'good' (black inside) and 'bad' (white inside) cells. Your task is to write an MCMC that automatically counts the number of good and bad cells, and displays the marginal posterior distribution over cell counts.



The image is a 100×100 pixel image, scaled between 0 (black) and 1 (white). You can download the array of numbers stored in the file `slide.mat` on the BUC5 website. Also there is the code (`makefake.m`) that produced this image.

As you can see, the 'true' image is defined by an ordered set of $r = (x, y)$ locations of cells (I have used pixel count as the units), with each location being either 'good' or 'bad'. You should model the *number* of cells as unknown, so random, with a Poisson distribution with some sensible mean (you can investigate this hyperparameter later). The *location* of each cell can be modelled as uniformly distributed within the image.

Write down the prior distribution that this model implies. By looking at the added noise term (in `makefake.m`), write down the likelihood function.

You can break the coding tasks into:

- Decide on a data structure for the state x. I suggest $x = \{n, (r_1, m_1), (r_2, m_2), \ldots, (r_n, m_n)\}$ with order setting back to front in the image.
- Calculate the prior for x (it's just the Poisson on n), $\pi(x)$.
- Simulate (noise free data) given x. That requires putting good and bad cells in the image.
- Evaluate the likelihood (from your simulated data), i.e., evaluate $\pi(y|x)$.

Having likelihhod and prior, you can now evaluate the posterior.

Now write a Metropolis-Hastings MCMC to explore the (un-normalized) posterior distribution. Your MCMC should utilize at least four *moves*:

1. A birth-death move that creates or deletes cells.
2. A swap move, that flips the label of a cell from 'good' to 'bad', and vice versa.
3. A translate move, that randomly translates the centre of a cell within a window.
4. A permute move that swaps the order (front to back) of a pair of cells.
5. Anything else you can think of that seems a good idea.

For each move, you will need to implement an appropriate accept-reject probability.

Plot the marginal posterior distribution over the number of good and bad cells.

(Challenge question – for the brave) Put a (hyper) prior distribution over the mean number of cells, and present results that integrate over this nuisance parameter. Does this make the counting more robust? (Simulate more data sets to test this in a frequentist sense.)

Colin Fox
23 Sep 2016