Project 5:
Object Detection
Gibbs and Metropolis Sampling

S M Al Mahi

Oklahoma State University

April 30, 2018

Project Objective

Objectives

1. Implement Jump Diffusion Markov Chain Monte Carlo(JD-MCMC) methods
2. Implemented Gibbs and Metropolis sampling
3. Apply them for object detection in image
4. Analyze the result with different parameter and test cases

Back Ground: MCMC

Objectives

1. MCMC is used for efficiently generate a set of discrete samples to approximate the underlying unknown distribution
2. Sample proposal is made
3. Sample is evaluated and then rejected or accepted based on a likelihood function

Back Ground: Model Order

Objectives

1. MCMC methods like Important sampling and Gibbs sampling assumes the model order to be known
2. In Object detection project we want to estimate model order K^*
3. JD-MCMC assumes the model order estimation as a Markov chain where states are the model orders
4. Where transition function is the ratio posterior probabilities of two neighboring state
5. The model order estimation is the stationary state of the Markov Chain

Back Ground: Model Order

Objectives

1. MCMC methods like Important sampling and Gibbs sampling assumes the model order to be known
2. In Object detection project we want to estimate model order K^*
3. JD-MCMC assumes the model order estimation as a Markov chain where states are the model orders
4. Where transition function is the ratio posterior probabilities of two neighboring state
5. The model order estimation is the stationary state of the Markov Chain

JD-MCMC: Algorithm

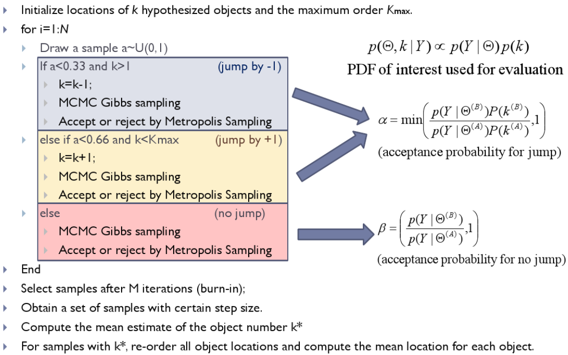

Figure: Random

JD-MCMC: Final Output

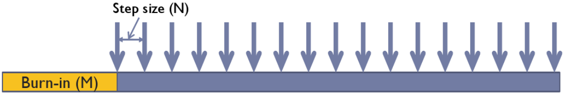

Figure: Random

Observations

- The JD-MCMC algorithm have two phases; Jump and Diffusion
- In the jump phase we induced prior knowledge about model order k^* using Poission distribution parameter λ
- We have run the experiment using good and bad priors by setting the λ
- The initial model order k^* is set as $k^* = \lambda + 2$
- We have found that all the test cases the object detection locks on all the objects correctly
- The model order also converges to the correct model order.
- We will show the result for some of the test cases throughout the following slides
- The Videos for objection and model order estimation processes have been attached with the slides

Test Case 1: Iteration N=50, Burn In M=10

ECEN-5283 Computer Vision

S M Al Mahi

- Project Objective
- Background
- Background
- Challenges and Caveats
- Effects of $\lambda andK*$
- Effects of $\lambda andK*$

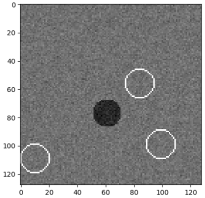

Figure: Initial

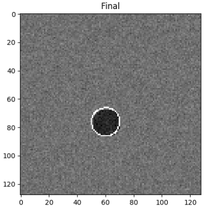

Figure: Final

Test Case 2: N=50, M=10

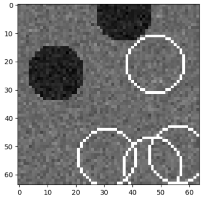

Figure: Initial

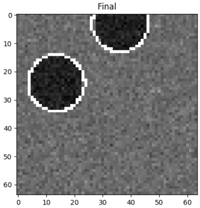

Figure: Final

Test Case 3: N=50, M=10

ECEN-5283 Computer Vision

S M Al Mahi

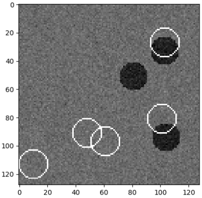

Figure: Initial

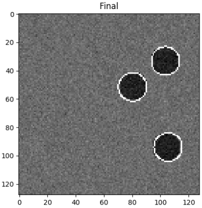

Figure: Final

Test Case 8: N=50, M=10

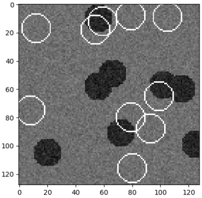

Figure: Initial

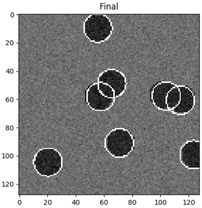

Figure: Final

Test Case 12: N=15, M=5

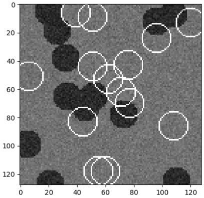

Figure: Initial

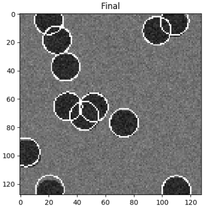

Figure: Final

Test Case 20: N=15, M=5

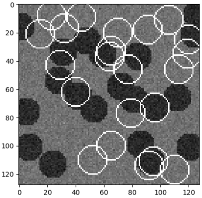

Figure: Initial

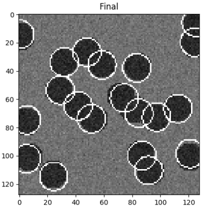

Figure: Final

Test Case 25: N=15, M=5

ECEN-5283 Computer Vision

S M Al Mahi

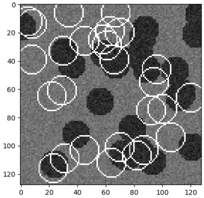

Figure: Initial

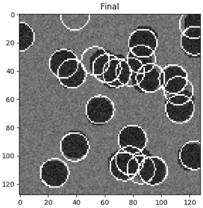

Figure: Final

Challenges and caveats

- The likelihood function has some limitations.
- It may produce greater likelihood for wrong model order than better detection with wrong model order.
- Thus the likelihood function is less efficient for model order estimation
- However, it is very useful for object detection. Thus in Gibbs Sampling we have used it
- For model order estimation the likelihood function should be multiplied by the prior for better model order estimation
- Because of the limitation of the likelihood function the prior is very crucial in model order estimation.
- Because of clip function object located at the corner and edge of the images are less likely to be detected using trivial approach.
- This problem was handled by changing the range for next move of an object using modified range on uniform distribution.

Limitation of Likelihood

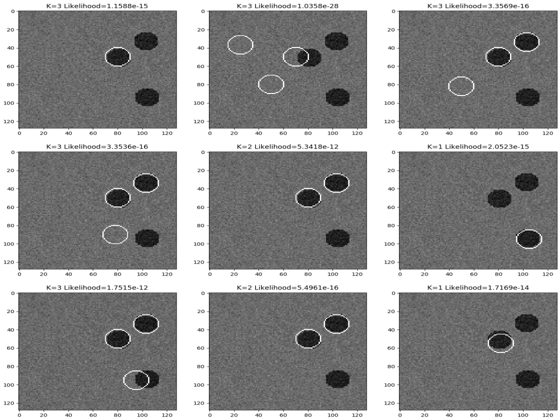

Figure: Even though the model order of the later one is correct the former produces higher likelihood.

Challenges and caveats

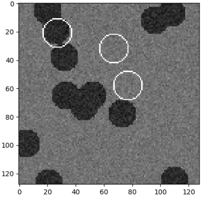

Figure: Initial

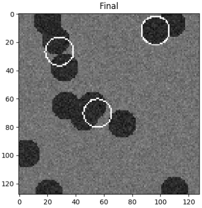

Figure: Final

Finally, we check the effect of choosing a bad $\lambda = 18, k^* = 3$ for disc 12 test case. It totally failed to converge to correct model order even after around double iterations.

Challenges and caveats

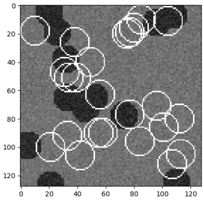

Figure: Initial

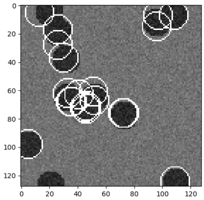

Figure: Final

Finally, we check the effect of choosing a bad $\lambda = 3, k^* = 25$ for disc 12 test case. It totally failed to converge to correct model order even after around double iterations. However, it locked on the objects correctly which reinforce our assumption that prior has a big role in model order estimation or the jump in JD-MCMC