Sampling

Sai Srinadhu Katta

1 Sampling

In this part of lab, density estimation using Gibbs Sampling is done. We are provided with Adult income dataset [**2**] as train data and test data. We are already provided with Bayes Net on the train data. Using this Bayes Net, Gibbs Sampler will generate samples, then for each data-point in test data probability with Bayes Net and probability from sample generation will be compared. Mean squared error is used as measure in all the below plots.

1.1 Gibbs Sampler

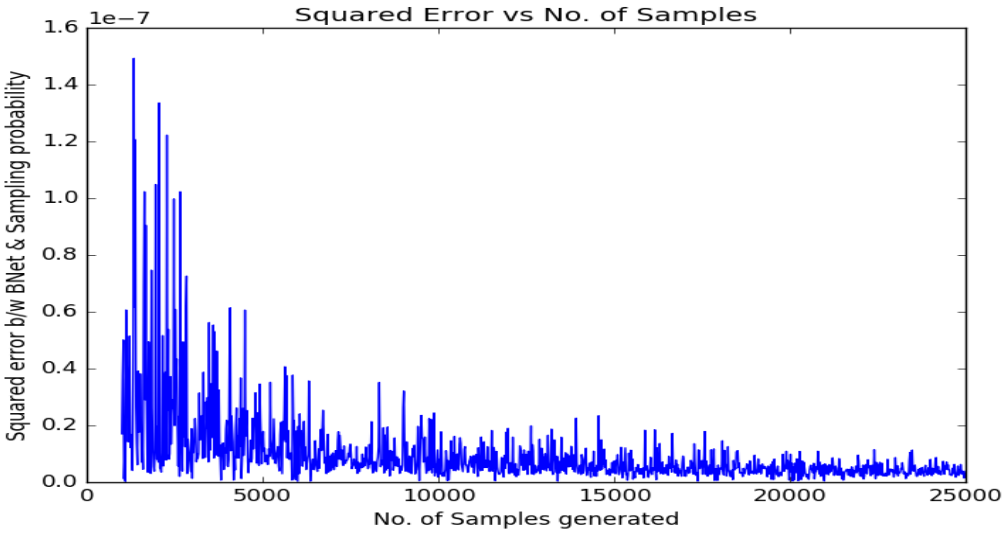
```
def Gibbs_Sampling(marginals, joints, trijoints, bayes_net_prob, test_samples, burn_in = 10, num_samples = 10000, num_features = 14,verbose = True):
    """
      Samples the given number of samples and returns the mean squared error from Bayes Net and Sampling for Test dataset. All about inputs and outputs is present in the code.
    """
    samples = [] #all the samples
    sample_init = sample_intilization(marginals, joints, trijoints, num_features = 14)
    samples.append(sample_init)
    sample = copy.deepcopy(sample_init)

    for i in range(num_samples): #get those many samples
        for j in range(burn_in): #wait till burn_in samples
            k = random.randrange(num_features)
            sample_init = sample_next_feature(marginals, joints, trijoints, sample, k)
            sample = copy.deepcopy(sample_init)

        #generate a sample and add it
        feature_index = random.randrange(num_features)
        sample_init=sample_next_feature(marginals,joints,trijoints,sample,feature_num=feature_index)

        sample = copy.deepcopy(sample_init)
        samples.append(sample)

        if (i%100 == 0 and verbose):
```

```
            print "iteration " + str(i) + " done."

    test_prob = Test_Sample_Prob(test_samples, samples)
    error_estimate  = Error_Sampling(bayes_net_prob, test_prob)

    return error_estimate
```

Code Snippet for Gibbs Sampler.

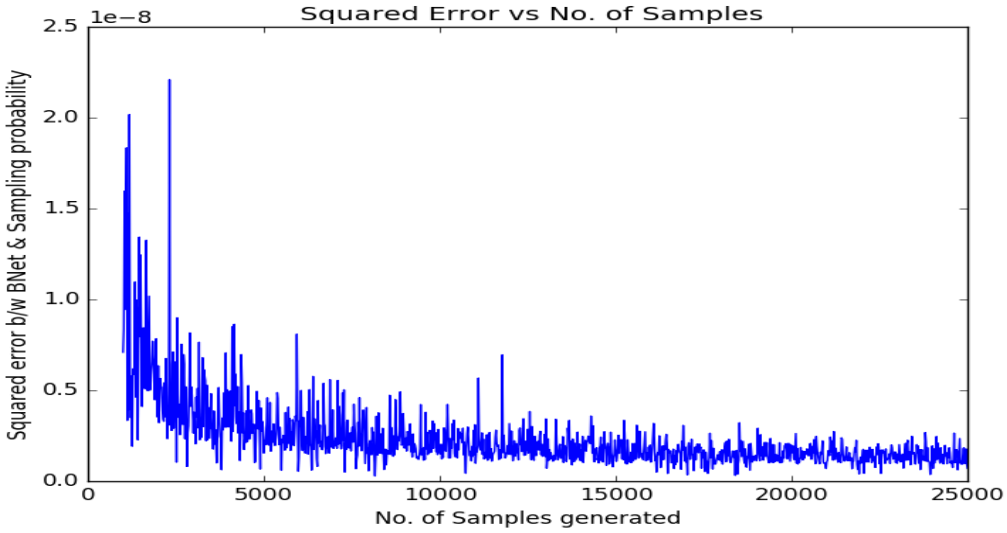
1.2 Plots & Observations

In the first plot burn_in set to 0, which means gibbs sampler's every sample will be considered in our sampled data, which may not be great since as such two samples picked one after another aren't truly independent. I started with 1000 samples and went till 25000 samples with a gap of 20. Some observations are that this plot is clearly more noisy compared to next one where we are using burn of 10 for which reason might be, not so much independence from samples. This has an advantage that this runs much faster compared to one which has burn since effective number of samples picked here is very less compared to one having burn. As the number of samples increases error decreases first and then almost goes to constant(decreases very less) which is intuitively expected.

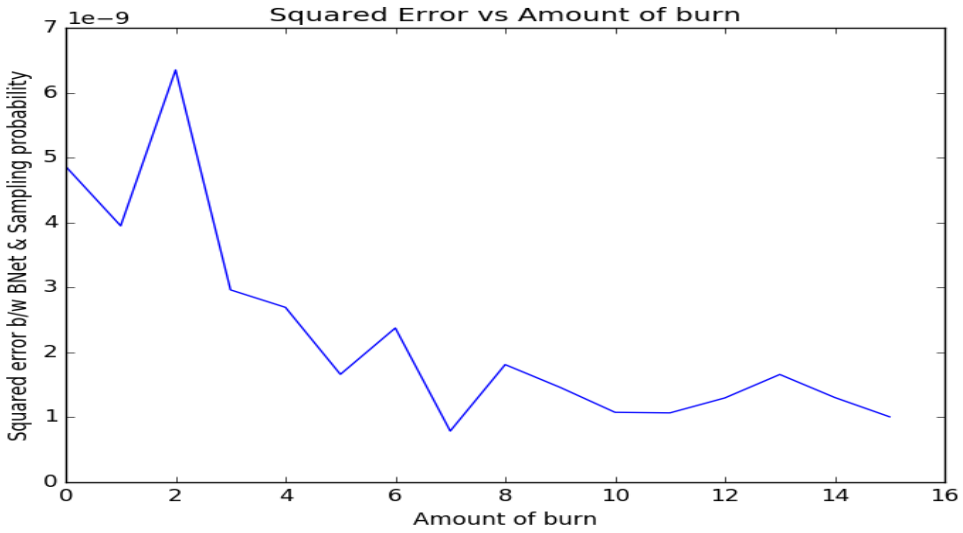


In the second plot burn_in set to 10, which means for gibbs sampler's every sample first burn_in number of samples are rejected and then next sample will be considered in our sampled data, which is better than without burn_in since samples now will be more independent or comes from many parts of joint probability space unlike without burn whose samples come from mostly localized spaces in joint probability distribution. I started with 1000 samples and went till 25000 samples with a gap of 20. Some observations are that this plot is clearly far less noisy compared to first one where we are not using burn for which reason might be, better independence of samples than without burn_in. This has an dis-advantage that

this runs much slower compared to one which has no burn since effective number of samples picked here is very huge. As the number of samples increases error decreases first and then almost goes to constant(decreases very less) which is intuitively expected.

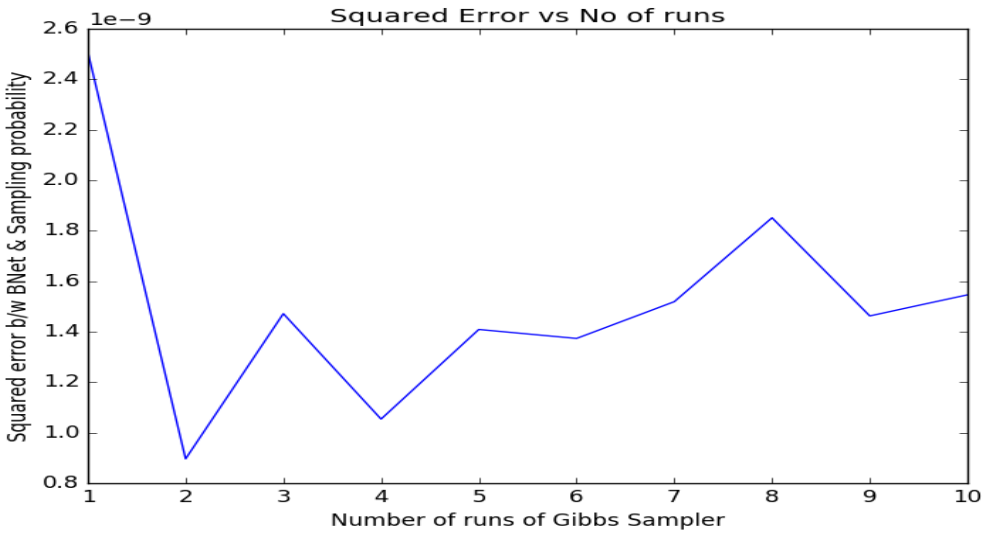


In the third plot burn_in is varied from 0 to 15 and 20000 samples are picked for each burn_in. The error measure is of mean squared. More the burn better it is, up's and down's can be due to stochastic behaviour of sampler and they aren't very drastic and so it's not a big concern for us. From the plot around burn_in of 10 seems to be great and so it is taken as burn_in in next plot. This also tells having a non-zero burn_in is usually better.



In the final plot burn_in is fixed to 10 and 20000 samples are picked. The number of runs of

sampler are varied from 1 to 10 and error is averaged out based on runs. The error measure is of mean squared. More the runs better it is, up's and down's can be due to stochastic behaviour of sampler and they aren't very drastic and so it's not a big concern for us. More times sampler is ran, errors tend to get averaged out and better estimate of errors comes for us.



The main observation from this is that for a good sampler fix a burn_in (say around 10) and fix number of samples as high as possible (say 20000), re-run the sampler as many times as possible (say around 5-10 times) and average out the error to get better and more accurate estimate of error on test data.

References

[1] LaTeX Templates for Laboratory Reports,
`https://github.com/mgius/calpoly_csc300_templates`

[2] Adult income dataset.
`https://archive.ics.uci.edu/ml/datasets/adult`