CP3 Official Solution by Course Staff

Collaboration Statement:

Mike Hughes prepared this solution, working alone.

Total hours spent: 4 hours

Links: [CP3 instructions] [Course collaboration policy]

Contents

1a: Figure

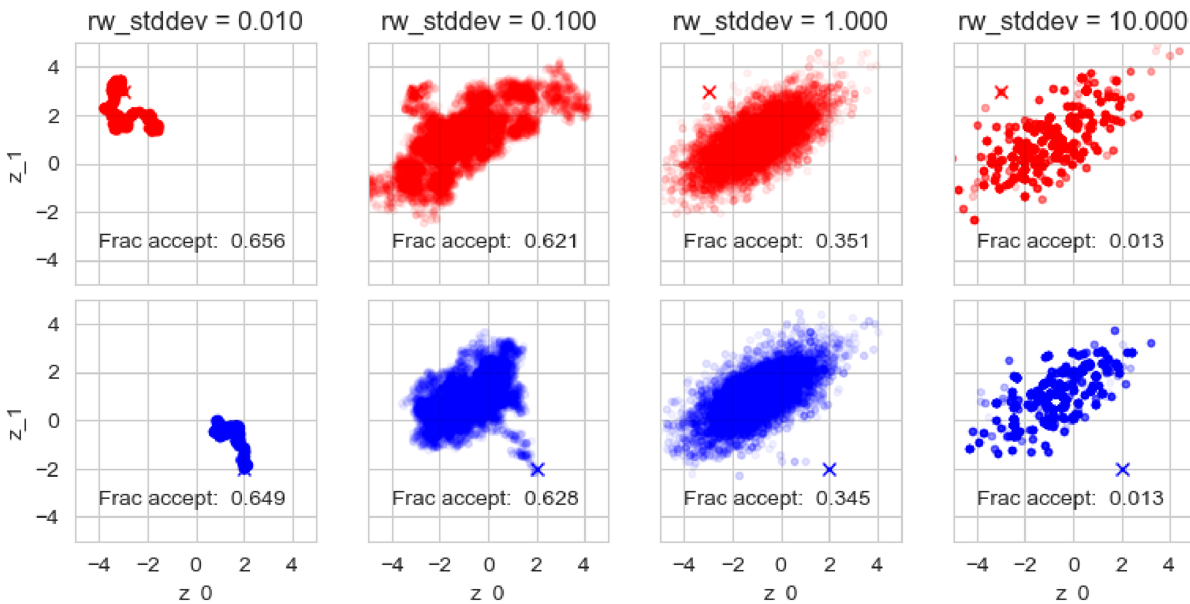
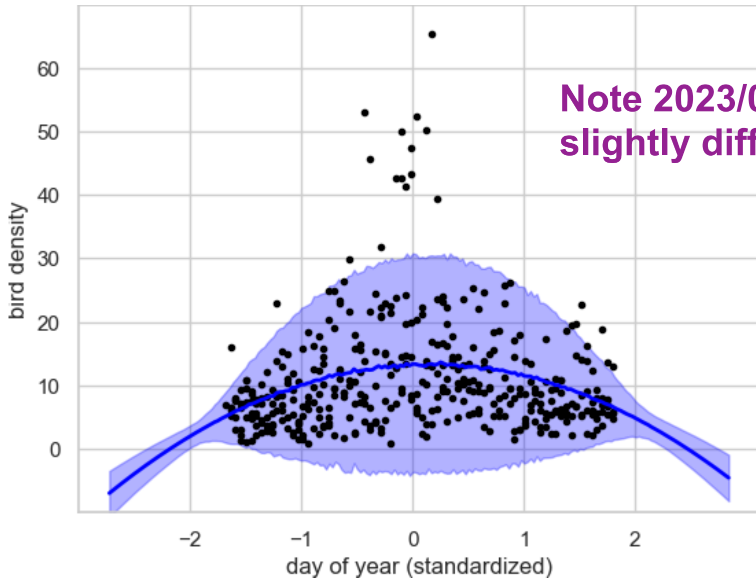


Fig. 1a: Behavior of RW Sampler from different proposal standard deviation values (columns) and initializations (rows, initial value marked with 'X'). Takeaway 1: Only $\sigma = 1.0$ seems to converge well. Smaller σ show that the Markov chain still depends on initial state. Larger σ values have too few accepted samples to yet be useful.

1b: Solution

No, I should always be skeptical that an MCMC chain has converged just from accept rate alone (or even just one chain alone). See figure 1a: high accept rate at low proposal width (small σ) means poor sampling from target.

2a: Solution



Note 2023/03/28:
slightly different plots possible

Fig. 2a: Predictive posterior for t_*: Both mean and variance can change as quadratic (order 2) functions of x_*.

2b: Solution

order	test score
0	-4.52 +/- 0.002
2	-4.04 +/- 0.005

Note 2023/03/28:
others may get slightly different results
we'll release more authoritative answers later after further investigation, based on student-submitted answers

2c: Solution

```
def calc_score(list_of_z_D, phi_RM, t_R):
    ''' Calculate per-example score averaged over provided test set of size R
    '''
    S = len(list_of_z_D)
    list_of_logpdf_R = []
    for ss in range(S):
        z_ss_D = list_of_z_D[ss]
        mean_R, stddev_R = unpack_mean_N_and_stddev_N(z_ss_D, phi_RM)
        logpdf_ss_R = scipy.stats.norm.logpdf(t_R, mean_R, stddev_R)
        list_of_logpdf_R.append(logpdf_ss_R)
    logpdf_SR = np.vstack(list_of_logpdf_R)
    logpdf_R = scipy.special.logsumexp(logpdf_SR, axis=0) - np.log(S)
    return np.mean(logpdf_R)
```