Simulation Results for Various Models

Multivariate Normal Inverse Gamma

For the multivariate normal - inverse gamma example, we consider the Bayesian regression setup.

$$y = X\beta + \varepsilon$$

where $y \in \mathbb{R}^N$, $X \in \mathbb{R}^{N \times p}$, $\beta \in \mathbb{R}^p$, and $\varepsilon \sim \mathcal{N}_N\left(0, \sigma^2 I_N\right)$.

The parameter $u = (\beta, \sigma^2) \sim \mathcal{NIG}(\cdot)$ is in $\mathbb{R}^D$.

For each D and for each N, the log marginal likelihood is calculated over 100 simulations. Each corresponding approximation uses 100 samples from the posterior.

For $D = 3, 5, 7, 10$, we compute the log marginal likelihood for different valuesof N. Since we can compute the true log marginal likelihood in closed form, we have a quantity with which we can compare the approximations generated by the tree-based algorithm. Results from the analysis are shown below. The true log marginal likelihood is denoted `LIL_N` and the corresponding approximation is directly beneath, denoted `LIL_N_hat`.

```
D = 3
              N = 50    N = 100   N = 150   N = 200   N = 300
LIL_N     -119.1375 -226.0264 -333.3248 -438.4928 -651.6707
LIL_N_hat -115.3618 -222.4969 -329.7781 -435.0519 -648.2589


D = 5
              N = 50    N = 100   N = 150   N = 200   N = 300
LIL_N     -122.9931 -230.1301 -336.8971 -443.2184 -655.1442
LIL_N_hat -115.7255 -223.3899 -329.9925 -436.6573 -648.5281


D = 7
              N = 50    N = 100   N = 150   N = 200   N = 300
LIL_N     -125.0980 -232.8142 -340.0919 -446.3879 -658.9896
LIL_N_hat -115.0527 -223.2259 -331.0643 -437.5655 -650.1652


D = 10
              N = 50    N = 100   N = 150   N = 200   N = 300
LIL_N     -149.3752 -263.4889 -374.5714 -483.5582 -697.5709
LIL_N_hat -136.1000 -251.1660 -363.0205 -472.1533 -686.4052
```

2-d Singular Example

Setup: $u \in [0,1]^2$, $K(u) = u_1^2 u_2^4$.

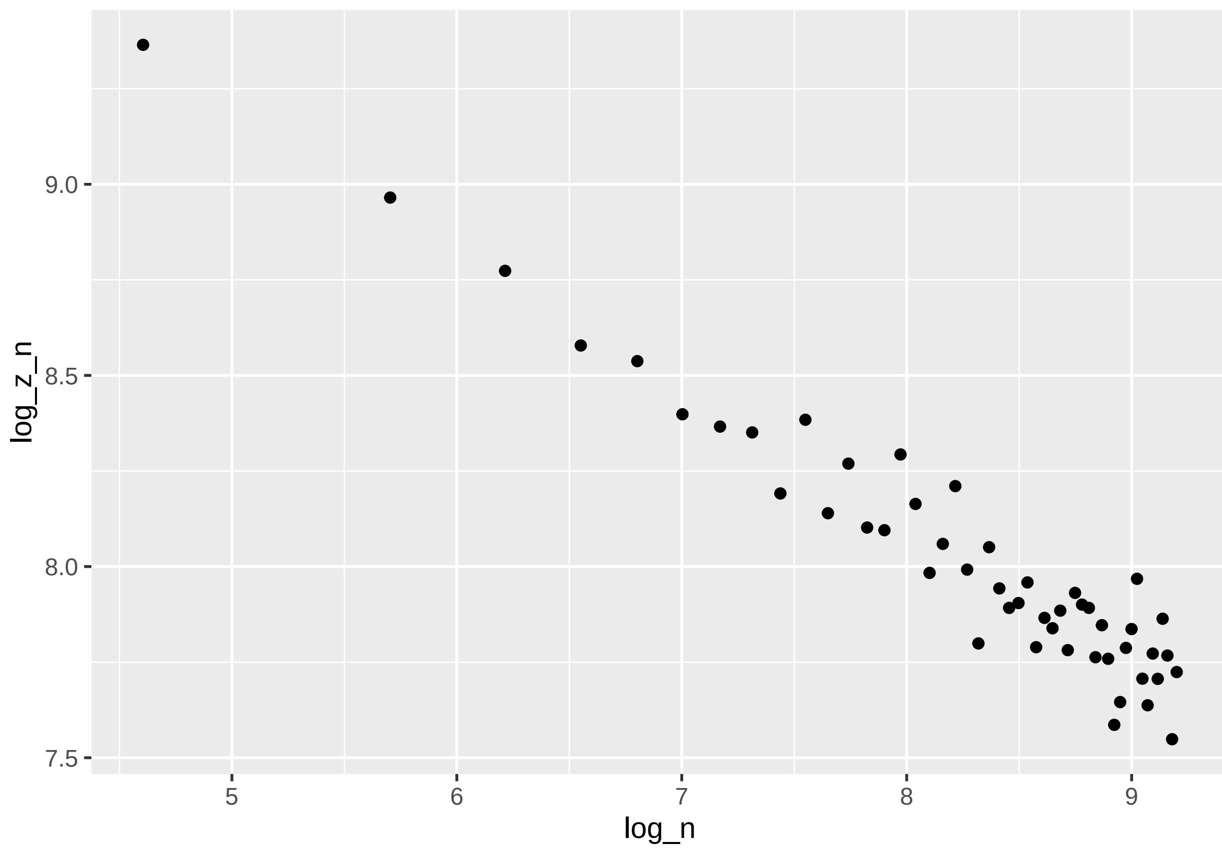
Density: $\gamma_n(u) = \frac{1}{\mathcal{Z}} \cdot e^{-nK(u)} \cdot \pi(u)$, where $\pi(\cdot)$ is the uniform measure on $[0,1]^2$.

We perform the following steps:

1. Use STAN to draw samples from γ_n for a given n
2. Use these samples to estimate $\mathcal{Z}_n$ using your method.
3. Repeat (1) and (2) for a range of values of n.
4. Regress your (estimate for) $\log \mathcal{Z}_n$ on $\log n$ and check if the slope parameter equals -0.25.

Simulation results below show a slope of -0.3562. This, however, is not the desired value. I'll look more closely at this and see what else can be done to improve the approximation.

Details: For each n, we averaged 200 approximations, with each approximation using $J = 2000$ samples from the $\gamma_N(\cdot)$




```
##
## Call:
## lm(formula = log_z_n ~ log_n, data = lil_df)
##
## Coefficients:
## (Intercept)        log_n
##     10.9585      -0.3562
```

Multivariate Skew Normal Example (varying N, D)

Following the approach in the notes and choosing

$$\psi(u) = \frac{1}{2}u'\Sigma u - \log \Phi(\alpha' u)$$

we know the normalizing constant is

$$\mathcal{Z} = (2\pi)^{D/2} \cdot \frac{1}{2}|\Sigma|^{1/2}$$

so the log normalizing constant is

$$\log \mathcal{Z} = \frac{D}{2}\log(2\pi) + \log\left(\frac{1}{2}\right) + \frac{1}{2}\log|\Sigma|$$

As written in the notes, we've taken $\Sigma = (D/N)\,\Omega$ for a fixed covariance matrix Ω.

In the following simulations, for each N, we approximate the log normalizing constant for $D = 2, \ldots, 10$. The true $\log \mathcal{Z}$ is denoted `log_Z` and its corresponding approximation is denoted `log_Z_hat`.

$N = 100$, and D ranging from 2 to 10,

Note approximations are quite poor for $D > 5$

```
                 D = 2     D = 3     D = 4     D = 5     D = 6
log_Z        -2.767293 -3.196168 -3.455145 -3.587785 -3.619748
log_Z_hat    -1.485009 -1.470474 -1.963887 -2.554357  1.729797

                 D = 7     D = 8      D = 9    D = 10
log_Z        -3.567988 -3.444553 -3.2584556 -3.016687
log_Z_hat     2.006608  1.682807  0.7620404  5.932742
```

$N = 200$, and D ranging from 2 to 10,

```
                 D = 2     D = 3     D = 4     D = 5     D = 6
log_Z        -3.460440 -4.235889 -4.841439 -5.320653 -5.699190
log_Z_hat    -3.625409 -2.782186 -4.299633 -3.519087 -3.750553

                 D = 7     D = 8     D = 9     D = 10
log_Z        -5.994003 -6.217142 -6.377618  -6.482423
log_Z_hat    -3.798406 -3.510086 -3.446793  -1.458410
```

$N = 500$, and D ranging from 2 to 10,

```
                 D = 2     D = 3     D = 4     D = 5      D = 6
log_Z        -4.376731 -5.610325 -6.674021 -7.611380  -8.448062
log_Z_hat    -3.763879 -5.099502 -5.042377 -6.326623  -5.305841

                 D = 7     D = 8      D = 9       D = 10
log_Z        -9.201020 -9.882305 -10.500926  -11.063877
log_Z_hat    -7.443714 -8.828195  -6.638853   -9.236426
```

$N = 1000$, and D ranging from 2 to 10,

```
              D = 2     D = 3     D = 4     D = 5      D = 6
log_Z     -5.069878 -6.650046 -8.060315  -9.344248 -10.52750
log_Z_hat -4.605873 -6.871010 -8.386169 -10.081417 -11.60774

              D = 7     D = 8     D = 9     D = 10
log_Z     -11.62704 -12.65489 -13.62009  -14.52961
log_Z_hat -13.84694 -15.92599 -16.65095  -16.93348
```

$N = 5000$, and D ranging from 2 to 10,

```
              D = 2     D = 3     D = 4     D = 5      D = 6
log_Z     -6.679316 -9.064203 -11.27919 -13.36784  -15.35582
log_Z_hat -6.885154 -9.646416 -10.19667 -13.77284  -17.10972

              D = 7     D = 8     D = 9     D = 10
log_Z     -17.26007 -19.09265 -20.86256  -22.57680
log_Z_hat -19.16342 -22.14646 -23.68828  -22.68788
```

Asymptotic Analysis

Regressing log marginal likelihood vs. log N

After taking your suggestions with regards to taking an average of log marginal likelihoods and subtracting the corresponding log marginal likelihood, I was able to get sensible results when regression the log marginal likelihood vs log N in the multivariate inverse gamma example.

Issues

Underflowing when for larger N

One quantity that became problematic to compute in some of the examples was $\psi(u)$ when the sample size increased beyond a couple hundred (in the case of the MVN-IG example). Since it's a function of the data, when we evaluate $\exp(\psi(u))$, this tends to underflow. I recall there being some numerical tricks that we can do in situations with underflow caused by the log likelihood, but since $\psi(u)$ is an explicit term in the approximation, I didn't think that directly playing with the quantities inside was a viable option.