tmvtnorm: A Package for the Truncated Multivariate Normal Distribution

by Stefan Wilhelm and B. G. Manjunath

Abstract   In this article we present **tmvtnorm**, an R package implementation for the truncated multivariate normal distribution. We consider random number generation with rejection and Gibbs sampling, computation of marginal densities as well as computation of the mean and covariance of the truncated variables. This contribution brings together latest research in this field and provides useful methods for both scholars and practitioners when working with truncated normal variables.

Introduction

The package **mvtnorm** is the first choice in R for dealing with the Multivariate Normal Distribution (Genz et al., 2009). However, frequently one or more variates in a multivariate normal setting $\mathbf{x} = (x_1, \ldots, x_m)^T$ are subject to one-sided or two-sided truncation ($a_i \leq x_i \leq b_i$). The package **tmvtnorm** (Wilhelm and Manjunath (2010)) is an extension of the **mvtnorm** package and provides methods necessary to work with the truncated multivariate normal case defined by $TN(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{a}, \mathbf{b})$.

The probability density function for $TN(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{a}, \mathbf{b})$ can be expressed as:

$$f(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{a}, \mathbf{b}) = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}}{\int_{\mathbf{a}}^{\mathbf{b}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\} d\mathbf{x}}$$

for $\mathbf{a} \leq \mathbf{x} \leq \mathbf{b}$ and 0 otherwise. We will use the following bivariate example with $\boldsymbol{\mu} = (0.5, 0.5)^T$ and covariance matrix $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 2 \end{pmatrix}$$

as well as lower and upper truncation points $\mathbf{a} = (-1, -\infty)^T, \mathbf{b} = (0.5, 4)^T$, i.e. x_1 is doubly, while x_2 is singly truncated. The setup in R code is:

```
> library(tmvtnorm)
> mu <- c(0.5, 0.5)
> sigma <- matrix(c(1, 0.8, 0.8, 2), 2, 2)
> a <- c(-1, -Inf)
> b <- c(0.5, 4)
```

The following tasks for dealing with the truncated multinormal distribution are described in this article:

- generation of random numbers

- computation of marginal densities

- computation of moments (mean vector, covariance matrix)

- estimation of parameters

Generation of random numbers

Rejection sampling

The first idea to generate variates from a truncated multivariate normal distribution is to draw from the untruncated distribution using `rmvnorm()` in the **mvtnorm** package and to accept only those samples inside the support region (i.e., rejection sampling). The different algorithms used to generate samples from the multivariate normal distribution have been presented for instance in Mi et al. (2009) and in Genz and Bretz (2009).

The following R code can be used to generate $N = 10000$ samples using rejection sampling:

```
> X <- rtmvnorm(n=10000, mean=mu,
>       sigma=sigma, lower=a, upper=b,
>       algorithm="rejection")
```

The parameter `algorithm="rejection"` is the default value and may be omitted.

This approach works well, if the acceptance rate α is reasonably high. The method `rtmvnorm()` first calculates the acceptance rate α and then iteratively generates N/α draws in order to get the desired number of remaining samples N. This typically needs just a few iterations and is fast. Figure 1 shows random samples obtained by rejection sampling. Accepted samples are marked as black, discarded samples are shown in gray. In the example, the acceptance rate is $\alpha = 0.432$.

```
> alpha <- pmvnorm(lower=a, upper=b, mean=mu,
>         sigma=sigma)
```

However, if only a small fraction of samples is accepted, as is the case in higher dimensions, the number of draws per sample can be quite substantial. Then rejection sampling becomes very inefficient and finally breaks down.

Gibbs sampling

The second approach to generating random samples is to use the Gibbs sampler, a Markov Chain Monte Carlo (MCMC) technique. The Gibbs sampler samples from conditional univariate distributions $f(x_i|\mathbf{x}_{-i}) = f(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$ which

are themselves truncated univariate normal distributions (see for instance Horrace (2005) and Kotecha and Djuric (1999)). We implemented the algorithm presented in the paper of Kotecha and Djuric (1999) in the **tmvtnorm** package with minor improvements. Using `algorithm="gibbs"`, users can sample with the Gibbs sampler.

```
> X <- rtmvnorm(n=10000, mean=mu,
>       sigma=sigma, lower=a, upper=b,
>       algorithm="gibbs")
```
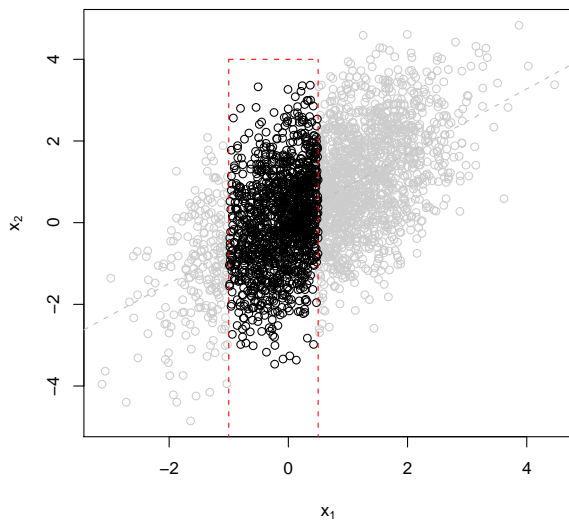

Figure 1: Random Samples from a truncated bivariate normal distribution generated by `rtmvnorm()`

Gibbs sampling produces a Markov chain which finally converges to a stationary target distribution. Generally, the convergence of MCMC has to be checked using diagnostic tools (see for instance the **coda** package from Plummer et al. (2009)). The starting value of the chain can be set as an optional parameter `start.value`. Like all MCMC methods, the first iterates of the chain will not have the exact target distribution and are strongly dependent on the start value. To reduce this kind of problem, the first iterations are considered as a burn-in period which a user might want to discard. Using `burn.in.samples=100`, one can drop the first 100 samples of a chain as burn-in samples.

The major advantage of Gibbs sampling is that it accepts all proposals and is therefore not affected by a poor acceptance rate α. As a major drawback, random samples produced by Gibbs sampling are not independent, but correlated. The degree of correlation depends on the covariance matrix Σ as well on the dimensionality and should be checked for using autocorrelation or cross-correlation plots (e.g. `acf()` or `ccf()` in the **stats** package).

```
> acf(X)
```

Taking only a nonsuccessive subsequence of the Markov chain, say every k-th sample, can greatly reduce the autocorrelation among random points as shown in the next code example. This technique called "thinning" is available via the `thinning=k` argument:

```
> X2 <- rtmvnorm(n=10000, mean=mu,
>       sigma=sigma, lower=a, upper=b,
>       algorithm="gibbs", burn.in.samples=100,
>       thinning = 5)
> acf(X2)
```

While the samples `X` generated by Gibbs sampling exhibit cross-correlation for both variates up to lag 1, this correlation vanished in `X2`.

Table 1 shows a comparison of rejection sampling and Gibbs sampling in terms of speed. Both algorithms scale with the number of samples N and the number of dimensions m, but Gibbs sampling is independent from the acceptance rate α and even works for very small α. On the other hand, the Gibbs sampler scales linearly with `thinning` and requires extensive loops which are slow in an interpreted language like R. From package version 0.8, we reimplemented the Gibbs sampler in Fortran. This compiled code is now competitive with rejection sampling. The deprecated R code is still accessible in the package via `algorithm="gibbsR"`.
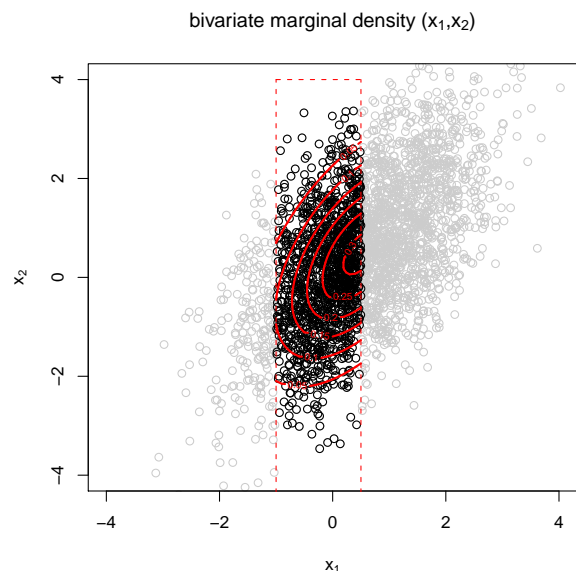

Figure 2: Contour plot for the bivariate truncated density function obtained by `dtmvnorm()`

Marginal densities

The joint density function $f(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{a}, \mathbf{b})$ for the truncated variables can be computed using `dtmvnorm()`. Figure 2 shows a contour plot of the bivariate density in our example.

Algorithm	Samples	$m = 2$		$m = 10$			
		$R_s = \prod_{i=1}^{2}(-1,1]$ $\alpha = 0.56$		$R_s = \prod_{i=1}^{10}(-1,1]$ $\alpha = 0.267$		$R_s = \prod_{i=1}^{10}(-4,-3]$ $\alpha = 5.6 \cdot 10^{-6}$	
Rejection Sampling	$N = 10000$	5600	0.19 sec	2670	0.66 sec	0	∞
	$N = 100000$	56000	1.89 sec	26700	5.56 sec	1	∞
Gibbs Sampling (R code, no thinning)	$N = 10000$	10000	0.75 sec	10000	3.47 sec	10000	3.44 sec
	$N = 100000$	100000	7.18 sec	100000	34.58 sec	100000	34.58 sec
Gibbs Sampling (Fortran code, no thinning)	$N = 10000$	10000	0.03 sec	10000	0.13 sec	10000	0.12 sec
	$N = 100000$	100000	0.22 sec	100000	1.25 sec	100000	1.21 sec
Gibbs Sampling (Fortran code and `thinning=10`)	$N = 10000$	10000	0.22 sec	10000	1.22 sec	10000	1.21 sec
	$N = 100000$	100000	2.19 sec	100000	12.24 sec	100000	12.19 sec

Table 1: Comparison of rejection sampling and Gibbs sampling for 2-dimensional and 10-dimensional TN with support $R_s = \prod_{i=1}^{2}(-1,1]$, $R_s = \prod_{i=1}^{10}(-1,1]$ and $R_s = \prod_{i=1}^{10}(-4,-3]$ respectively: number of generated samples, acceptance rate α and computation time measured on an Intel® Core™ Duo CPU@2.67 GHz

However, more often marginal densities of one or two variables will be of interest to users. For multinormal variates the marginal distributions are again normal. This property does not hold for the truncated case. The marginals of truncated multinormal variates are not truncated normal in general. An explicit formula for calculating the one-dimensional marginal density $f(x_i)$ is given in the paper of Cartinhour (1990). We have implemented this algorithm in the method `dtmvnorm.marginal()` which, for convenience, is wrapped in `dtmvnorm()` via a `margin=i` argument. Figure 3 shows the Kernel density estimate and the one-dimensional marginal for both x_1 and x_2 calculated using the `dtmvnorm(...,  margin=i) i = 1,2` call:

```
> x   <- seq(-1, 0.5, by=0.1)
> fx <- dtmvnorm(x, mu, sigma,
>       lower=a, upper=b, margin=1)
```

The bivariate marginal density $f(x_q, x_r)$ for x_q and x_r ($m > 2, q \neq r$) is implemented based on the works of Leppard and Tallis (1989) and Manjunath and Wilhelm (2009) in method `dtmvnorm.marginal2()` which, again for convenience, is just as good as `dtmvnorm(..., margin=c(q,r))`:

```
> fx <- dtmvnorm(x=c(0.5, 1),
>       mu, sigma,
>       lower=a, upper=b, margin=c(1,2))
```

The help page for this function in the package contains an example of how to produce a density contour plot for a trivariate setting similar to the one shown in figure 2.

Computation of moments

The computation of the first and second moments (mean vector $\mu^* = E[\mathbf{x}]$ and covariance matrix Σ^* respectively) is not trivial for the truncated case, since they are obviously not the same as μ and Σ from the parametrization of $TN(\mu, \Sigma, \mathbf{a}, \mathbf{b})$. The moment-generating function has been given first by Tallis (1961), but his formulas treated only the special case of a singly-truncated distribution with a correlation matrix $\mathbf{R}$. Later works, especially Lee (1979) generalized the computation of moments for a covariance matrix Σ and gave recurrence relations between moments, but did not give formulas for the moments of the double-truncated case. We presented the computation of moments for the general double-truncated case in a working paper Manjunath and Wilhelm (2009) and implemented the algorithm in the method `mtmvnorm()`, which can be used as

```
> moments <- mtmvnorm(mean=mu, sigma=sigma,
>           lower=a, upper=b)
```

The moment calculation for our example results in $\mu^* = (-0.122, 0)^T$ and covariance matrix

$$\Sigma^* = \begin{pmatrix} 0.165 & 0.131 \\ 0.131 & 1.458 \end{pmatrix}$$

To compare these results with the sample moments, use

```
> colMeans(X)
> cov(X)
```

It can be seen in this example that truncation can significantly reduce the variance and change the covariance between variables.

Estimation

Unlike our example, in many settings μ and Σ are unknown and must be estimated from the data. Estimation of (μ, Σ) when truncation points $\mathbf{a}$ and $\mathbf{b}$ are known can typically be done by either Maximum-Likelihood, Instrumental Variables (Amemiya (1973), Amemiya (1974), Lee (1979), Lee (1983)) or in a Bayesian context (Griffiths (2002)). We are planning to include these estimation approaches in the package in future releases.
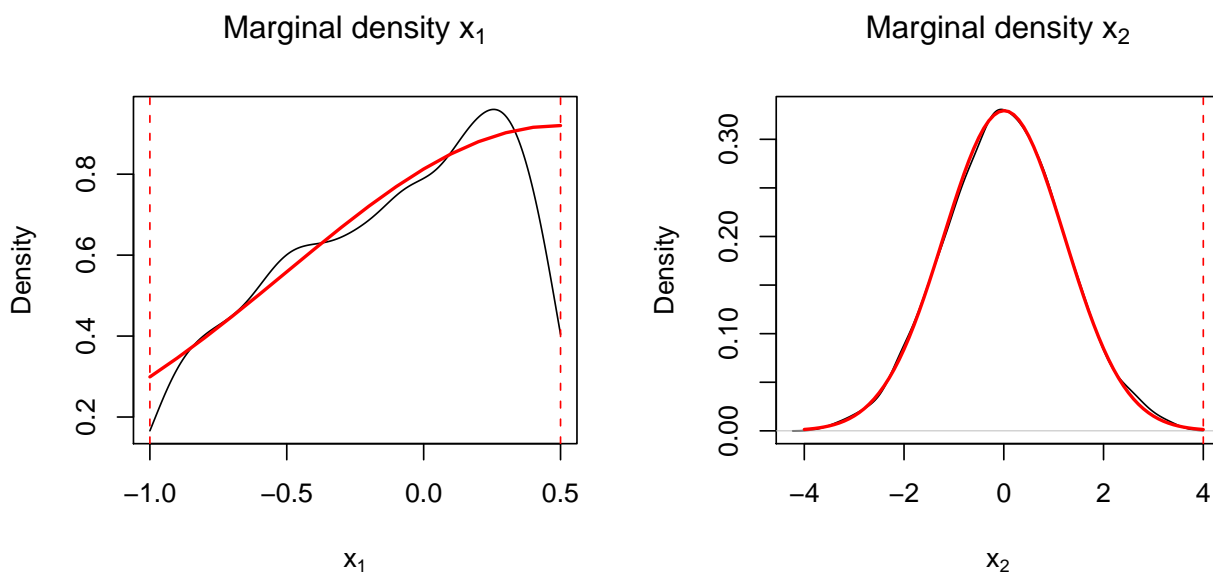
Figure 3: Marginal densities for x_1 and x_2 obtained from Kernel density estimation from random samples generated by `rtmvnorm()` and from direct calculation using `dtmvnorm(...,margin=1)` and `dtmvnorm(...,margin=2)` respectively

Summary

The package **tmvtnorm** provides methods for simulating from truncated multinormal distributions (rejection and Gibbs sampling), calculations from marginal densities and also calculation of mean and covariance of the truncated variables. We hope that many useRs will find this package useful when dealing with truncated normal variables.

Bibliography

T. Amemiya. Regression analysis when the dependent variable is truncated normal. *Econometrica*, 41: 997–1016, 1973.

T. Amemiya. Multivariate regression and simultaneous equations models when the dependent variables are truncated normal. *Econometrica*, 42:999–1012, 1974.

J. Cartinhour. One-dimensional marginal density functions of a truncated multivariate normal density function. *Communications in Statistics - Theory and Methods*, 19:197–203, 1990.

A. Genz and F. Bretz. *Computation of Multivariate Normal and t Probabilities*, volume 195 of *Lecture Notes in Statistics*. Springer-Verlag, Heidelberg, 2009.

A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2009. URL http://CRAN.R-project. org/package=mvtnorm. R package version 0.9-8.

W. Griffiths. A Gibbs' sampler for the parameters of a truncated multivariate normal distribution. 2002. URL http://ideas.repec.org/p/mlb/ wpaper/856.html.

W. C. Horrace. Some results on the multivariate truncated normal distribution. *Journal of Multivariate Analysis*, 94:209–221, 2005.

J. H. Kotecha and P. M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. *IEEE Computer Society*, pages 1757–1760, 1999. http://dx.doi.org/ 10.1109/ICASSP.1999.756335.

L.-F. Lee. On the first and second moments of the truncated multi-normal distribution and a simple estimator. *Economics Letters*, 3:165–169, 1979.

L.-F. Lee. The determination of moments of the doubly truncated multivariate tobit model. *Economics Letters*, 11:245–250, 1983.

P. Leppard and G. M. Tallis. Evaluation of the mean and covariance of the truncated multinormal. *Applied Statistics*, 38:543–553, 1989.

B. G. Manjunath and S. Wilhelm. Moments calculation for the double truncated multivariate normal density. Working Paper, 2009. URL http: //ssrn.com/abstract=1472153.

X. Mi, T. Miwa, and T. Hothorn. mvtnorm: New numerical algorithm for multivariate normal probabilities. *R Journal*, 1:37–39, 2009.

M. Plummer, N. Best, K. Cowles, and K. Vines. *coda: Output analysis and diagnostics for MCMC*, 2009. R package version 0.13-4.

G. M. Tallis. The moment generating function of the truncated multinormal distribution. *Journal of the Royal Statistical Society, Series B*, 23(1):223–229, 1961.

S. Wilhelm and B. G. Manjunath. *tmvtnorm: Truncated Multivariate Normal Distribution*, 2010. URL http://CRAN.R-project.org/package=tmvtnorm. R package version 0.9-2.

Stefan Wilhelm
Department of Finance
University of Basel, Switzerland
wilhelm@financial.com

B. G. Manjunath
Department of Mathematics
University of Siegen, Germany
bgmanjunath@gmail.com