Scalable Bayesian dynamic covariance modeling with variational Wishart and inverse Wishart processes

Creighton Heaukulani (No affiliation) and Mark van der Wilk (PROWLER.io)

Summary

- Apply black-box, gradient-based variational inference to Wishart and inverse Wishart processes
- Introduce a numerically stable additive noise parameterization and a low-rank "factored" variant
- May scale down inference w.r.t. length of time series and dimensionality of covariance matrix
- Competitive performance against MGARCH, though inverse Wishart process appears unreliable

Construct (inverse) Wishart processes from GPs

Let $Y_n \in \mathbb{R}^D$ be the n-the measurement in a time series of length N, and let

$$Y_n \mid \mu_n, \Sigma_n \sim \mathcal{N}(\mu_n, \Sigma_n), \qquad n \geq 1. \tag{1}$$

Here we focus on modeling the sequence of covariance matrices Σ_n.

Consider the **Wishart** and **inverse Wishart processes**, which may be constructed from i.i.d. Gaussian processes as follows: Sample a bunch of i.i.d. GPs:

$$f_{d,k} \sim \mathrm{GP}(0, \kappa(\,\cdot\,, \cdot\,; \theta)), \qquad d \leq D,\, k \leq \nu, \tag{2}$$

for some *degrees of freedom* $\nu \geq D$. Let X_n be a covariate representing the time at which measurement n was taken, and let $F_{n,d,k} := f_{d,k}(X_n)$, let $F_n := (F_{n,d,k}, d \leq D, k \leq \nu)$, and

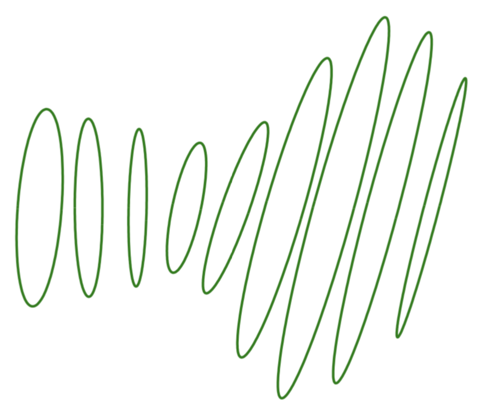
Construct a Wishart process: $\Sigma_n = AF_nF_n^TA^T$, (3)

Or an inverse Wishart process: $\Sigma_n^{-1} = AF_nF_n^TA^T$, (4)

for a *scale matrix* A.

Intuitively, we can visualize an evolving sequence of two-dimensional covariance matrices as in the figure to the right.

* Figure taken from Wilson and Ghahramani [2010].



Black-box variational inference with GPflow

We maximize the lower bound on $\log p(Y)$:

$$\log p(Y) \geq \sum_{n=1}^{N} \mathbb{E}_{q(F_n)}[\log p(Y_n \mid F_n)] - \sum_{d=1}^{D}\sum_{k=1}^{\nu} \mathrm{KL}[q(U_{d,k}) \,||\, p(U_{d,k})], \tag{5}$$

for a mean-field variational distribution q, specified by Hensman et al. [2015]. Gradients obtained following Kingma and Welling [2014], Salimans and Knowles [2013]; black-box requiring only:

$$\log p(Y_n \mid F_n) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|AF_nF_n^TA^T| - \frac{1}{2}Y_n^T(AF_nF_n^TA^T)^{-1}Y_n, \tag{6}$$

for the Wishart process, or for the inverse Wishart process:

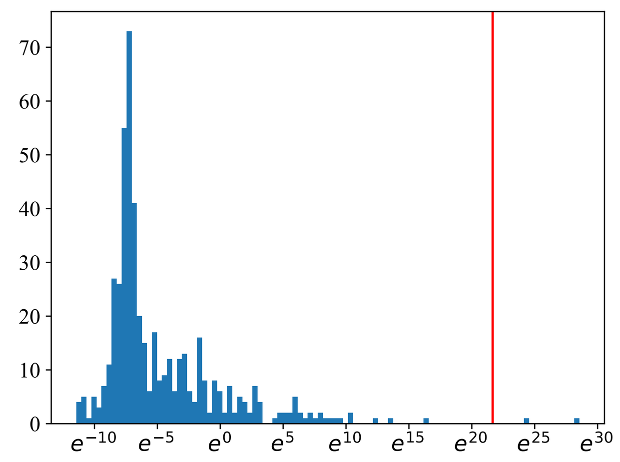
$$\log p(Y_n \mid F_n) = -\frac{D}{2}\log(2\pi) + \frac{1}{2}\log|AF_nF_n^TA^T| - \frac{1}{2}Y_n^TAF_nF_n^TA^TY_n. \tag{7}$$

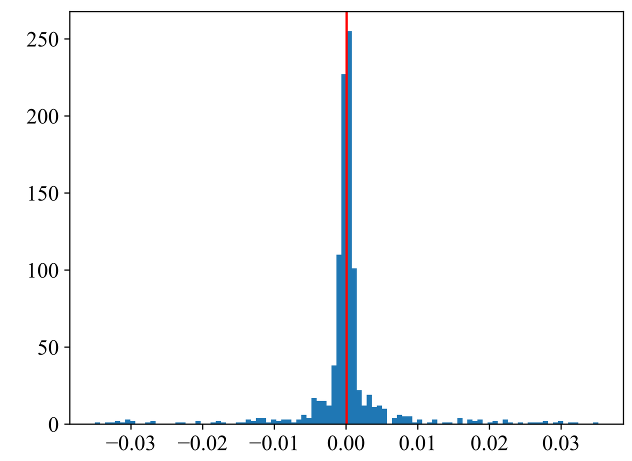
Particularly easy with GPflow [Matthews et al., 2017]. Code for the inverse Wishart process:

```
import numpy as np
import tensorflow as tf
from gpflow import models, likelihoods, kernels, conditionals, kullback_leiblers, transforms

class InvWishartProcessLikelihood(likelihoods.Likelihood):
    def __init__(self, D, R=1):
        super().__init__()
        self.R, self.D = R, D
        self.A_diag = Parameter(np.ones(D), transform=transforms.positive)

    @gpflow.decors.params_as_tensors  # decorator translating TF tensors for GPflow
    def variational_expectations(self, mu, S, Y):
        N, D = tf.shape(Y)
        W = tf.random_normal([self.R, N, tf.shape(mu)[1]])
        F = W * (S ** 0.5) + mu  # samples through which TF automatically differentiates

        # compute the (mean of the) likelihood
        AF = self.A_diag[:, None] * tf.reshape(F, [self.R, N, D, -1])
        yffy = tf.reduce_sum(tf.einsum('jk,ijkl->ijl', Y, AF) ** 2.0, axis=-1)
        chols = tf.cholesky(tf.matmul(AF, AF, transpose_b=True))  # cholesky of precision
        logp = tf.reduce_sum(tf.log(tf.matrix_diag_part(chols)), axis=2) - 0.5 * yffy
        return tf.reduce_mean(logp, axis=0)

class InvWishartProcess(models.svgp.SVGP):
    def __init__(self, X, Y, Z, minibatch_size=None, nu=None):
        D = Y.shape[1]
        nu = D if nu is None else nu  # degrees of freedom

        # create a compositional kernel function
        kern = kernels.Matern32(1) + kernels.RationalQuadratic(1) + kernels.PeriodicKernel(1) * kernels.RBF(1)

        # almost all work is done by SVGP!
        super().__init__(X, Y, kern = kern,  # notation as in the paper
                         likelihood = InvWishartProcessLikelihood(D, R=10),  # 10 MCMC samples
                         num_latent = D * nu,  # number of outputs (multi-output GP)
                         minibatch_size = minibatch_size,
                         Z = Z)  # initial inducing points should be passed
```

The additive noise and factored parameterizations

Inference on the Wishart process is unstable. The LHS shows a histogram of one of the gradients.





Correct unstable gradients in the Wishart process with the following additive noise parameterization:

$$\Sigma_n := AF_nF_n^TA^T + \Lambda, \quad n \geq 1, \tag{8}$$

and the loglikelihood becomes:

$$\log p(Y_n \mid F_n) = \frac{ND}{2}\log(2\pi) - \log|AF_nF_n^TA^T + \Lambda| - \frac{1}{2}Y_n^T(AF_nF_n^TA^T + \Lambda)^{-1}Y_n. \tag{9}$$

Corrected gradients in the RHS figure above.
An analogous parameterization for the inverse Wishart process is:

$$\Sigma_n^{-1} := AF_nF_n^TA^T + \Lambda^{-1}, \quad n \geq 1, \tag{10}$$

with loglikelihood:

$$\log p(Y_n \mid F_n) = \frac{ND}{2}\log(2\pi) + \log|AF_nF_n^TA^T + \Lambda^{-1}| - \frac{1}{2}Y_n^T(AF_nF_n^TA^T + \Lambda^{-1})Y_n. \tag{11}$$

Set $K \ll D$, and let F_n be $K \times \nu$ for $\nu \geq K$, and let A be $D \times K$. We obtain a low-rank, "factored" model. With the Woodbury matrix identities, the computational complexity becomes:

	Time	Space
full cov ($M \ll N$)	$O(N_bD^3 + D^2(N_b^3 + N_bM^2 + M^3))$	$O(D^2(N_b^2 + N_bM + M^2))$
factored cov ($K \ll D$)	$O(N_bDK^2 + K^2(N_b^3 + N_bM^2 + M^3))$	$O(DK + K^2(N_b^2 + N_bM + M^2))$

Comparison to MGARCH

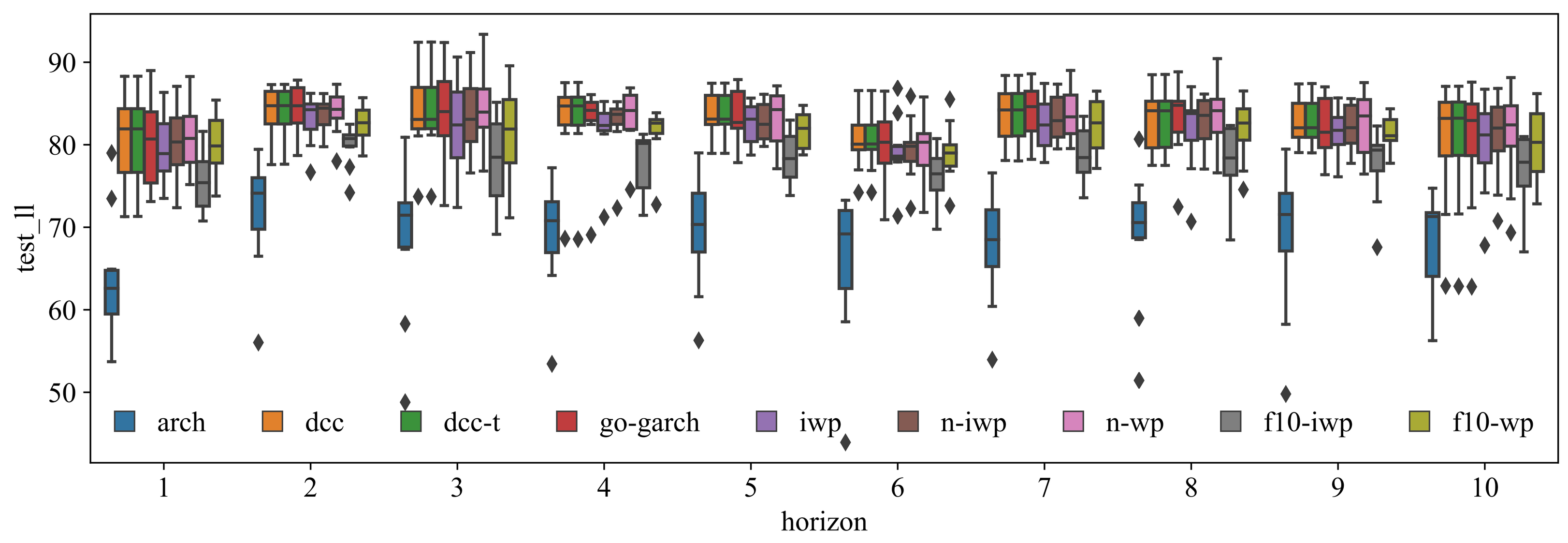
Summary:

- Inverse Wishart process is unreliable.
- Additive noise parameterization improves performance.
- Should prefer full covariance model if resources allow.

Datasets:

- **Dow 30**: Intraday returns on the comps of the Dow 30 Industrial Ave; $N = 978$, $D = 30$.
- **FX**: Daily foreign exchange rates for 20 currency pairs; $N = 1,565$, $D = 20$.
- **ETF**: Close prices of 52 popular exchange traded funds; $N = 2,285$ and $D = 52$.
- **S&P 500**: Daily returns on closing prices of comps of the S&P 500 index; $N = 1,258$, $D = 505$.

We evaluate on test sets of 10 forecast horizons. Visualize as follows:



The boxplots are over 10 training/testing splits, formed with rolling windows.

	Dow 30	FX	S&P 500
arch	142.47 ± 17.97	68.24 ± 7.55	1358.23 ± 355.12
dcc	162.70 ± 42.98	82.52 ± 4.55	–
dcc-t	162.64 ± 42.86	82.54 ± 4.56	–
go-garch	163.59 ± 52.65	82.43 ± 4.85	–
iwp	164.09 ± 26.47* (1.71e-8)	81.42 ± 4.12 (8.15e-8)	–
n-iwp	164.49 ± 19.82* (1.13e-9)	82.10 ± 3.72 (1.62e-3)	–
n-wp	$\mathbf{165.98 \pm 23.23}$* (1.03e-6)	$\mathbf{82.69 \pm 4.15}$ (5.11e-2)	–
f10-iwp	162.28 ± 22.91* (4.67e-11)	77.76 ± 3.94 (5.99e-17)	1275.27 ± 264.48 (4.14e-18)
f10-wp	165.39 ± 30.89* (2.31e-5)	81.12 ± 3.59 (2.62e-10)	$\mathbf{1423.31 \pm 132.19}$* (1.48e-13)
f30-iwp	–	–	1047.73 ± 1436.16 (3.90e-18)
f30-wp	–	–	$\mathbf{1438.40 \pm 130.14}$* (1.54e-15)

References

J. Hensman, A. G. de G. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *AISTATS*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

A. G. de G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017.

T. Salimans and D. A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

A. G. Wilson and Z. Ghahramani. Generalised Wishart processes. In *UAI*, 2010.