

# Multi-task GP

Thursday, March 28, 2024 11:45 AM

[GPflow/doc/source/notebooks/advanced/varying\\_noise.ipynb at develop-1.0 · GPflow/GPflow · GitHub](https://GPflow.readthedocs.io/en/develop/source/notebooks/advanced/varying_noise.ipynb)

Both the original implementation by Bonilla in Matlab and the Gpytorch implementation are using same input

[https://proceedings.neurips.cc/paper\\_files/paper/2007/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2007/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf). Multi-task Gaussian Process Prediction (Edwin V. Bonilla, Kian Ming A. Chai, Christopher K.I. Williams)

We propose a model that learns a shared covariance function on input-dependent features and a “free-form” covariance matrix over tasks.

We propose a model that attempts to learn inter-task dependencies based solely on the *task identities* and the observed data for each task.

In our model, this is achieved by having a common covariance function over the features  $x$  of the input observations. This contrasts with the semiparametric latent factor model [5] where, with the same set of input observations, one has to estimate the parameters of several covariance functions belonging to different latent processes.

[https://docs.gpytorch.ai/en/stable/examples/03\\_Multitask\\_Exact\\_GPs/Multitask\\_GP\\_Regression.html](https://docs.gpytorch.ai/en/stable/examples/03_Multitask_Exact_GPs/Multitask_GP_Regression.html)

% Edwin V. Bonilla (edwin.bonilla@nicta.com.au)  
% Last update: 23/01/2011

[https://github.com/ebonilla/mtgp/blob/master/utils/generate\\_data.m](https://github.com/ebonilla/mtgp/blob/master/utils/generate_data.m)

[https://github.com/ebonilla/mtgp/blob/master/scripts/alpha\\_mtg.m](https://github.com/ebonilla/mtgp/blob/master/scripts/alpha_mtg.m)

<https://github.com/cornellius-gp/gpytorch/issues/1743>  
[Question] Implementing multi-output multi-task approximate GP #1743

<https://github.com/choltz95/MTGP-NN/tree/master>

Coregionalized regression with Gpy

## Multiple Output Gaussian Processes

- [Coregionalization with Gaussian Processes](#) This tutorial shows the use of a coregionalized model within GPy. In particular such models can be used for multi-task or multi-output learning.

- **Coregionalization on Marathon Data** This tutorial runs the multioutput regression on a higher level, introducing stacked hierarchical multitask regression.

<https://nbviewer.org/github/SheffieldML/notebook/blob/master/GPy/multiple%20outputs.ipynb>

[https://bigaidream.gitbooks.io/subsets\\_ml\\_cookbook/content/bayes/gp/coregionalized\\_regression\\_gpy.html](https://bigaidream.gitbooks.io/subsets_ml_cookbook/content/bayes/gp/coregionalized_regression_gpy.html)

[https://github.com/SheffieldML/notebook/blob/deploy/GPy/multiple\\_outputs/Introduction%20to%20Multiple%20Output%20Gaussian%20Processes.ipynb](https://github.com/SheffieldML/notebook/blob/deploy/GPy/multiple_outputs/Introduction%20to%20Multiple%20Output%20Gaussian%20Processes.ipynb)

[https://notebook.community/SheffieldML/notebook/GPy/coregionalized\\_regression\\_tutorial](https://notebook.community/SheffieldML/notebook/GPy/coregionalized_regression_tutorial)

<https://github.com/SheffieldML/notebook/blob/master/GPy/multiple%20outputs.ipynb>

<https://stackoverflow.com/questions/61475796/using-gpy-multiple-output-coregionalized-prediction>

[https://gpy.readthedocs.io/en/devel/\\_modules/GPy/kern/src/coregionalize.html](https://gpy.readthedocs.io/en/devel/_modules/GPy/kern/src/coregionalize.html)

[https://gpy.readthedocs.io/en/deploy/\\_modules/GPy/models/sparse\\_gp\\_coregionalized\\_regression.html](https://gpy.readthedocs.io/en/deploy/_modules/GPy/models/sparse_gp_coregionalized_regression.html)

<https://mlatcl.github.io/gpss/lectures/05-multi-output-gps.html>

Gpytorch

[https://docs.gpytorch.ai/en/v1.6.0/\\_modules/gpytorch/kernels/multitask\\_kernel.html](https://docs.gpytorch.ai/en/v1.6.0/_modules/gpytorch/kernels/multitask_kernel.html)

Multi-task kernels

[https://docs.gpytorch.ai/en/v1.6.0/\\_modules/gpytorch/kernels/multitask\\_kernel.html](https://docs.gpytorch.ai/en/v1.6.0/_modules/gpytorch/kernels/multitask_kernel.html)

using `:class:`gpytorch.kernels.IndexKernel`` as a basic multitask kernel.

[https://github.com/cornellius-gp/gpytorch/blob/master/gpytorch/kernels/index\\_kernel.py](https://github.com/cornellius-gp/gpytorch/blob/master/gpytorch/kernels/index_kernel.py)  
 rank (int):

Rank of  $:math:`B`$  matrix. Controls the degree of correlation between the outputs. With a **rank of 1** the outputs are identical except for a scaling factor.

[https://nbviewer.org/github/SheffieldML/notebook/blob/master/GPy/coregionalized\\_regression](https://nbviewer.org/github/SheffieldML/notebook/blob/master/GPy/coregionalized_regression)

## [tutorial.ipynb](#)

The covariance kernel for multiple output is:

KroneckerProduct(B,K), where  $B = WW^T + \kappa I$ . K is kernel function for input X, B is the coregionalization matrix, and should be positive definite.

The following notebook talks about the impact of rank in details

[https://github.com/SheffieldML/notebook/blob/deploy/GPy/multiple\\_outputs/Introduction%20to%20Multiple%20Output%20Gaussian%20Processes.ipynb](https://github.com/SheffieldML/notebook/blob/deploy/GPy/multiple_outputs/Introduction%20to%20Multiple%20Output%20Gaussian%20Processes.ipynb)

The problem with this model is that it is quite restrictive, we're assuming that all the functions are simply scaled instances of the *same* latent function. Note that we also have a diagonal matrix added, thus our coregionalisation matrix B is already full-rank (a rank one matrix, i.e. , would have a correlation of one).

We can extend our model; imagine instead that our outputs are computed by the weighted sum of **two** latent functions, described (at the moment) with the same kernel. This leads to our matrix being of rank 2:

The practical upshot is that far more variation can be expressed between the outputs, and therefore more of a prediction is likely to be based on the values of the other inputs. This can cause overfitting too. It also means there's now more parameters to fit during hyperparameter optimization.

Botorch

[https://botorch.org/v/0.1.0/api\\_modules/botorch/models/multitask.html](https://botorch.org/v/0.1.0/api_modules/botorch/models/multitask.html)

04/10/2024 implementation note

Redo uncertainty figure (generate more points?)

LMGP: dimensions of output variables

If stratify by task ind, task 3(group 3) would have much better performance. If don't stratify, group 5 would have better performance. In general, if not stratify, the chart looks better

If stratify by features only one datapoint of PS90 and Sucrose are greater than 0

[https://docs.gpytorch.ai/en/latest/examples/03\\_Multitask\\_Exact\\_GPs/Hadamard\\_Multitask\\_GP\\_Regression.html](https://docs.gpytorch.ai/en/latest/examples/03_Multitask_Exact_GPs/Hadamard_Multitask_GP_Regression.html)

## Hadamard Multitask GP Regression

Here, we assume that we have observations for **one task per input**. For each input, we specify the task of the input that we observe. (The kernel that we learn is expressed as a Hadamard product of an input kernel and a task kernel)

*Multiply the two together to get the covariance we want*

covar=covar\_x.mul(covar\_i)

[https://en.wikipedia.org/wiki/Hadamard\\_product\\_\(matrices\)](https://en.wikipedia.org/wiki/Hadamard_product_(matrices))

Hierarchical in Na\_CL  
Na\_CL added or not  
If added, how much

More realistic uncertainty graph  
Add a balance line

Covariance matrix dimension  
Diagram for Multi-task

Other multitask GP implementations

[https://botorch.org/api/\\_modules/botorch/models/multitask.html](https://botorch.org/api/_modules/botorch/models/multitask.html)

<https://notebook.community/jrg365/gpytorch/examples/03>

[Multitask GP Regression/Multitask GP Regression Scalable With KISSGP](#)

<https://discourse.mc-stan.org/t/modelling-dependent-time-series-with-gaussian-processes/4265>

<https://www.ijcai.org/Proceedings/11/Papers/238.pdf>

Multi-Kernel Gaussian Processes

<https://discourse.pymc.io/t/coregionalization-model-for-two-separable-multidimensional-gaussian-process/2550>

<https://ieeexplore.ieee.org/document/9746570>

## Multitask Gaussian Process With Hierarchical Latent Interactions

<https://eprints.whiterose.ac.uk/114503/1/1106.6251v2.pdf>

!!!Alvarez, M.A., Rosasco, L. and Lawrence, N.D. orcid.org/0000-0001-9258-1030 (2012) Kernels for Vector-Valued Functions: a Review. *Foundations and Trends® in Machine Learning*, 4 (3). pp. 195-266. ISSN 1935-8237

<https://github.com/cornellius-gp/gpytorch/issues/2072>

[Bug] Multi-input Multi-output GP with Variational Inference reporting Incorrect Variances # 2072