This document illustrates the usage of the program package for developing the marginalized particle filter (MPF)-based recursive multiple-output sparse Gaussian Processes inference and regression. The following example can reproduce some results for the "Synthetic Data" experiment (second toy example) in the paper "Online Sparse Multi-Output Gaussian Process Regression and Learning" by Le Yang, Ke Wang and L. S. Mihaylova, which is going to appear in 2019 in *IEEE Trans. Signal and Information Processing over Networks.*

**Main.m**

This is the main program.

**toy\_example2\_data.m**

function [training,test,v] = toy\_example2\_data(K\_training,N\_training,M,K\_test,sigma)

This function generates the training and test datasets, as well as the inducing point locations for the second toy example.

The inputs are:

K\_training Number of training datasets,

N\_training Number of data points in each training dataset,

M Number of inducing points,

K\_test Number of test data points,

sigma Noise standard deviation for the test data.

The outputs are:

training Training datasets,

test Test dataset,

v Inducing point locations.

Two well-correlated bivariate functions:

f1(x1, x2) = 3cos(x1) + 4cos(2x2)

f2(x1, x2) = 2cos(x1) + 3cos(2x2)

are considered in this example.

The locations of the inducing points of each latent sparse GP are randomly selected from the region corresponding to the Cartesian product [-4:5; 4:5]\*[-4:5; 4:5].

**model\_init.m**

function par = model\_init(P,Q,D,v,M,meanfun,covfun)

This function initializes the multiple-output sparse Gaussian process model.

The inputs are:

P Number of model outputs,

Q Number of sparse Gaussian processes,

D Input dimension,

v Locations of inducing points,

M Number of inducing points,

meanfun Mean function,

covfun Covariance function.

The output is:

par Initialized model parameters.

The output ‘par’ is a data structure which includes:

par.P Number of model outputs,

par.Q Number of sparse GPs,

par.D Input dimension,

par.beta Output noise precisions,

par.w Mixing coefficient matrix,

par.g.mhyp Initialled hyperparameters for the mean functions,

par.g.m Mean of GPs,

par.g.loghyp Initialled hyperparameters for the covariance functions,

par.g.C Covariance of GPs(jittered),

par.g.invC Inverse covariance of GPs,

par.v Locations of inducing points,

par.M Number of inducing points,

par.meanfun Mean function,

par.covfun Covariance function.

**ELBO.m**

function elbo = ELBO(y,par1,par2,HN,P,R,bCg,x)

This function evaluates the evidence lower bound of the log marginal of the current training data set.

The inputs are:

y Measurement vector,

par1 Original model parameters,

par2 Updated model parameters,

HN Measurement matrix,

P Projection matrix,

R Measurement noise covariance,

bCg Conditional prediction covariance of latent variables,

x Input vectors.

The output is:

elbo Evidence lower bound.

**marginalized\_PF.m**

function particles = marginalized\_PF(particles,elbo)

This function realizes the marginalized particle filter-based hyperparameter learning for multiple-output Gaussian process regression and learning.

The inputs are:

particles Input particles,

elbo Evidence lower bound (in logarithm).

The output are:

particles Updated particles.

The structure ‘particles’ includes:

particles.par Model parameters,

particles.weight Particle weight (in logarithm).

**model\_learning.m**

function par = model\_learning(particles,training,R,xmean,xstd,ymean,ystd)

This function realizes the learning of the multiple-output sparse Gaussian processes model using a marginalized particle filter.

The inputs are:

particles Input particles,

training Training dataset,

R Starting point (number of training datasets used for initialization).

The output is:

par Trained model parameters.

**model\_pred.m**

function [par,y,Cy,Cvy,bCg,HN,P,R] = model\_pred(par,x,yt)

This function produces the predicted output and its covariance for the multiple-output sparse Gaussian Processes model.

The inputs are:

par Model parameters,

x Input vectors.

The outputs are:

par Updated model parameters,

y Predicted model output,

Cy Covariance of model output,

Cvy Cross covariance,

bCg Conditional prediction covariance of latent variables,

HN Mixing coefficient matrix for inputs in x,

P Projection matrix,

R Measurement noise covariance.

**model\_regression.m**

function [par,elbo] = model\_regression(par,x,y)

This function finds the variational approximation of the posterior mean and covariance of the inducing points for the multiple-output sparse Gaussian processes model.

The inputs are:

par Model parameters,

x Input vectors,

y Measurement vectors.

The outputs are:

par Updated model parameters,

elbo Evidence lower bound (in logarithm).

**particle\_init.m**

function [par,xmean,xstd,ymean,ystd] = particle\_init(P,Q,D,v,M,meanfun,covfun,training,R)

This function initializes the multiple-output sparse Gaussian process model using Nyugen's code (stochastic variational inference (SVI)).

The inputs are:

P Number of model outputs,

Q Number of sparse Gaussian processes,

D Input dimension,

v Locations of inducing points,

M Number of inducing points,

meanfun Mean function,

covfun Covariance function,

training Training datasets,

R Number of training datasets used for initialization.

The output are:

par Initialized model parameters,

xmean Mean of the input,

xstd Standard deviation of the input,

ymean Mean of the output,

ystd Standard deviation of the output.