wrapper creation tutorial

Creating a wrapper for a new GP package

This vignette serves as a tutorial on how to create a new wrapper for a GP package by editing the file `WrappedGP.R`, turning it into the wrapper for a new GP package. The file is essentially a copy of the file `WrappedmlegpGP.R`, a wrapper for the `mlegp` package. We assume that the user has some knowledge of GP implementations in R.
We require the following methods for the new GP package:

- Create a new GP

- Make a prediction for a new data point with a given GP create by the package, returning the mean and the error

- Store the covariance function of the GP

First, we describe a minimum version which goes over all necessary changes. Afterwards, we point the user to more advanced implementations which can be found in `WrappedDiceKrigingGP.R`.

Changes in `CreateWrappedGP.R`

- Change the name of the package in `CreateWrappedGP.R` by replacing "dummy" with the name of the package

Changes in `WrappedGP.r`

- Change `gp_control` in line 109. Here, we mean all arguments which are **not** the design matrix texttt{X}, the target variable `y` or it's uncertainties `y_var`. These include e.g. the covariance function, parameters for the optimizer, etc.

- Update the method `update_init_covpars` in l.170. In a `gp` from `mlegp`, the length scales of each dimension are stored under `gp$beta`. The standard deviation of the whole GP is stored under `gp$sig2`, and the (constant) mean under `gp$mu[1]`.

- Update the method `get_lengthscales` in l. 180. Use the same location that you used for the length scales in `update_init_covpars`.

- Update the method `get_cov_mat` in l.224. Replace the argument of the return function in l. 229 with the location of the covariance function in the `gp` object.

- Update the method `call_create_gp` in l. 317. Start by replacing the method which creates the GP `mlegp::mlegp` in l.321 with the one of the new package. Update the control parameters, too.

- Update the method `call_predict` in l. 356. Start by replacing the method `mlegp::predict.gp` and it's arguments used to define `predictions` in l. 358 and 366. Then, replace the locations of the mean and standard error in the `prediction` in l. 363 and 371. In `mlegp`, the mean is stored under `prediction$fit`, and the standard error under `prediction$se.fit`.

- Update the method `predict` in l. 432. This step is only necessary if `add_buffer_in_prediction = TRUE`. Update the temporary gp `temp_gp` in l. 461 by changing the method for gp creation `mlegp::mlegp` and the control parameters.

Optional and advanced features

Subsequent features are not necessary for the proper function of the package. Since these aspects are highly dependent on the chosen package, we merely make the user aware of these features and point them to the code sections in `WrappedDiceKrigingGP.R`.

- Including the prediction uncertainty. This can simply achieved by effectively replacing `X, y` in l. 322 and 462 in `WrappedGP.r` with `X, y, y_var`, assuming that this is the proper order of the arguments in the new package.

- Including a retrain buffer. Here, the train function needs to be adapted. See the beginning of the `train` method in l. 352 - 367 in `WrappedDiceKrigingGP.R`.

- Create a GP with given covariance parameters. If the package allows it, the GP parameters can be set by the field `init_covpars`. This can be seen in `WrappedDiceKrigingGP.R` in l. 562 - 576.

- Use bounds for GP parameters (i.e. length scales). This can be seen in `WrappedDiceKrigingGP.R` in l. 484 - 507.