PBL

```
PBL (Preference-Based Learning) is the main class for the POLAR (Preference Optimization and
Learning Algorithm for Robotics) toolbox. This class includes the
posterior sampling algorithms for regret minimization and active
learning

@author Maegan Tucker @date 2020-11-27

Copyright (c) 2020, AMBER Lab All right reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted only in compliance with the BSD 3-Clause
license, see http://www.opensource.org/licenses/bsd-license.php
```

Class Details

Superclasses	handle
Sealed	false
Construct on load	false

Constructor Summary

PBL	POLAR class constructor

Property Summary

comp_time	structure containing information regarding computation time
feedback	structure with compiled user/simulated feedback
iteration	structure with information specific to each iteration
metrics	evaluation metrics
post_model	structure containing posterior model updated at each iteration
previous_data	structure containing information loaded from previous experimental data
sample_table	history of number of sampled actions in terms of
settings	structure of learning and action space settings
unique_visited_action_globalInds	global indices corresponding to unique_visited_actions
unique_visited_actions	list of uniquely visited actions
unique_visited_isCoac	flags corresponding to if visited actions were sampled or coactive

Method Summary

addFeedback	Uses user/simulated feedback to update posterior
addPreviousData	Constructs obj.previous_data
addToHistory	if new iteration, append row to history
addlistener	Add listener for event.
algSetup	Constructs obj.settings
appendVisitedInd	appends action and corresponding index to obj.unique_visited_actions,
delete	Delete a handle object.
eq	== (EQ) Test handle equality.
findobj	Find objects matching specified conditions.
findprop	Find property of MATLAB handle object.
ge	>= (GE) Greater than or equal relation for handles.

	getGlobalInd	Gets global index of action according to obj.settings.points_to_sample
	getLinearSubspace	
	getMapping	
	getNewActions	Draws new actions to query using @Sampling class
	getVisitedInd	gets unique index of action based on obj.unique_visited_actions
	gt	> (GT) Greater than relation for handles.
Sealed	isvalid	Test handle validity.
	le	<= (LE) Less than or equal relation for handles.
	listener	Add listener for event without binding the listener to the source object.
	lt	< (LT) Less than relation for handles.
	ne	~= (NE) Not equal relation for handles.
	notify	Notify listeners of event.
	postProcess	
	removeLargeMatrices	Delete large matrices stored in post_model (to save on computation time)
	reset	
	runExperiment	Begin a new experiment using the settings loaded in obj.settings
	runSimulation	Begin a new simulation using the settings loaded in obj.settings
	testLengthscales	Generates a set of 1D objective functions by sampling from the GP prior.
	updateBestAction	
	updatePosterior	Updates posterior over points_to_include which are updated to

Event Summary

ObjectBeingDestroyed	Notifies listeners that a particular object has been destroyed.