

sample2.c	main.py	Settings	report1.tex	resourcesNewVersio...
313	# note that all learned data are represented by color gray with different marker t			
314	pl.plot(class1.toNdarray[:,0], class1.toNdarray[:,1], 'o', color='tab:gray', label			
315	pl.plot(class1.mean[0], class1.mean[1], 'o', color='k', markersize=8)			
316	pl.plot(class2.toNdarray[:,0], class2.toNdarray[:,1], '+', color='tab:gray', label			
317	pl.plot(class2.mean[0], class2.mean[1], '+', color='k', markersize=10)			
318				
319	# MARK: - plot test data distinguished by the correct class of each point. (given			
320	# note that all test points are plotted with color blue.			
321	testPointsInClass1 = testData.toNdarray[nu.array(result)==1]			
322	testPointsInClass2 = testData.toNdarray[nu.array(result)==2]			
323	pl.plot(testPointsInClass1[:,0], testPointsInClass1[:,1], 'o', color='tab:blue', l			
324	pl.plot(testPointsInClass2[:,0], testPointsInClass2[:,1], '+', color='tab:blue', l			
325				
326	# plot the recognition line (if exists)			
327	x,y = sy.symbols('x,y')			
328	if type(recogLine) is RecogLine: # for euclideanDistanceMethod, weightDistanceMeth			
329	poly = recogLine.polyExpr			
330	xValues = nu.arange(-10, 80, 10)			
331	yValues = [sy.solve(poly.subs({x: value}), y) for value in xValues]			
332	pl.plot(xValues, yValues, color='r')			
333				
334	elif type(recogLine) is list: # for nearestNeighborMethod			
335	rawValues = [point.rawValue for point in recogLine]			
336	xValues, yValues = [value[0] for value in rawValues], [value[1] for value i			
337	pl.plot(xValues, yValues, '.', color='r', markersize=1)			
338				
339	elif type(recogLine) is Eigen: # for 主成分分析			
340	for vector in recogLine.vectors:			
341	midlePoint = (class1.mean + class2.mean) / 2.0			
342	poly = vector[1,0]/vector[0,0] * (x - midlePoint[0,0]) + midlePoint[1,0]			
343	xValues = nu.arange(-10, 80, 10)			
344	yValues = [poly.subs({x: value}) for value in xValues]			
345	pl.plot(xValues, yValues, color='r')			
346				
347	pl.axis(plotAxisDict[DATASET_ID])			
348	pl.legend()			
349	pl.title(title)			
350	pl.savefig('results/' + title + figureFile)			
351	pl.figure()			
352				