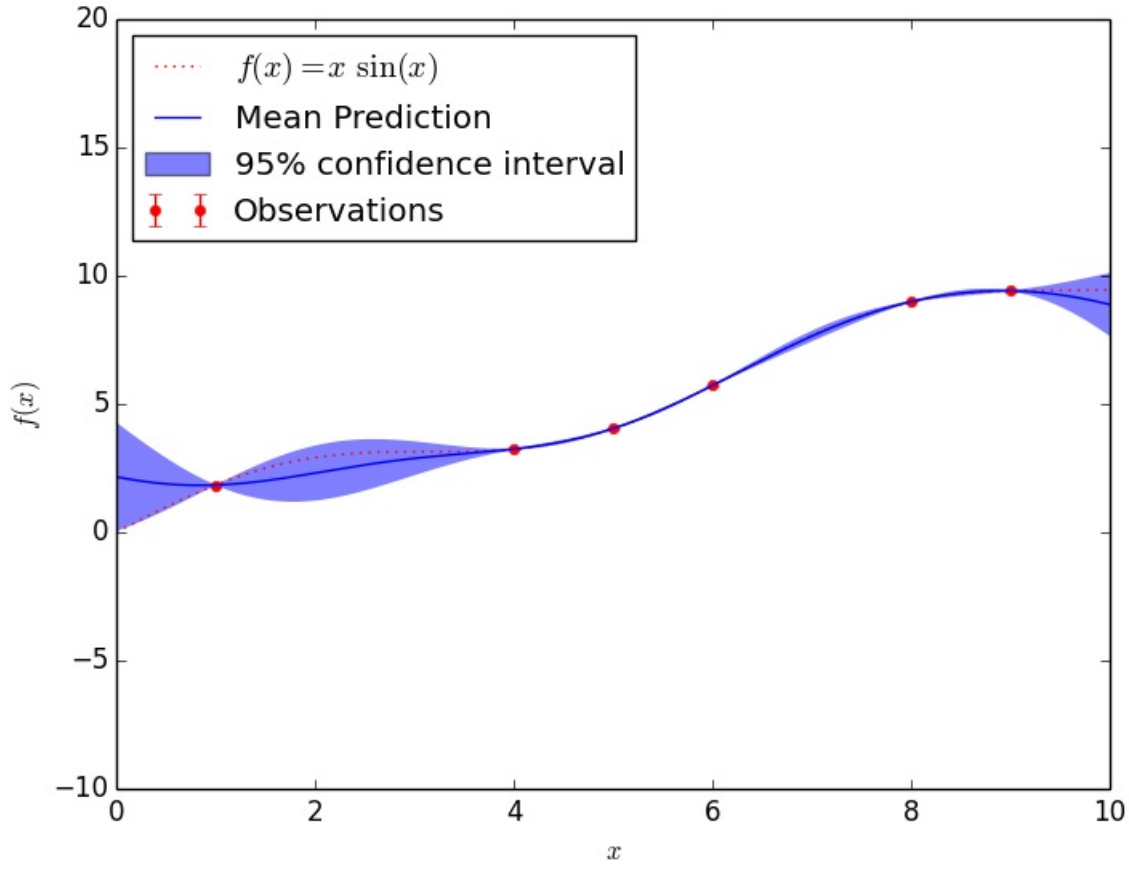
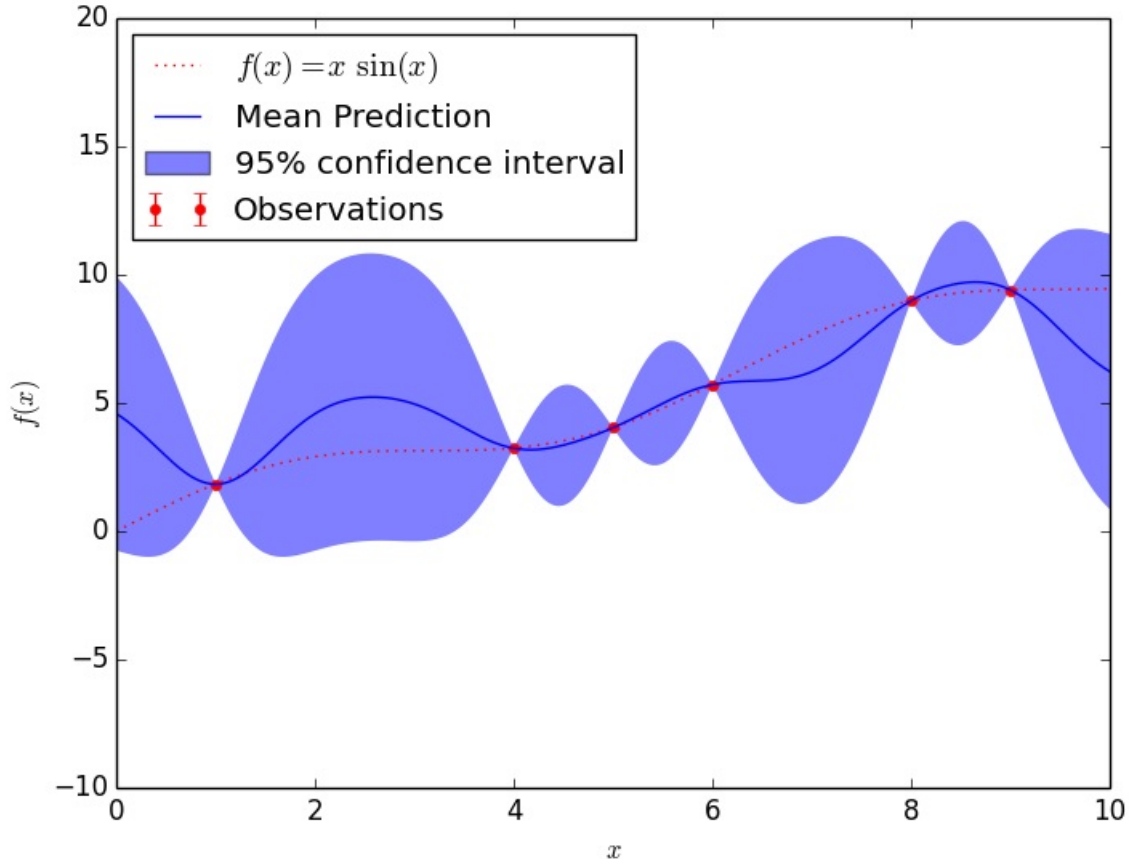
1. Guassian Process Regression



\



1.
2. The variance expands
3. 'squared_exponential', 'generalized_exponential', 'absolute_exponential', 'linear', 'cubic'

4. generalized_exponential

2. Bayesian Optimazation

1. from 'x=np.linspace(-2,2,200)'\
2. It is in /black_box/objectives.py
3. GP is trained in optimize()
4. (1) Generate training data
   (2) Define objective function
   (3) Training Gaussian process in interations: finding a valuable training point based on Expected Improvement

3.

```
 X = self._ensure_shape(X)
y_pred, y_variance = self.gp.predict(X, eval_MSE=True)
 if predict_variance:
   return y_pred, y_variance
 else:
     return y_pred
```

4.

```
 X = self._ensure_shape(X)
Y_pred, Y_variance = self.gp.predict(X, eval_MSE=True)
y_plus = np.max(Y_pred)
result = np.zeros(Y_pred.shape)
Xi = 0.1
for i in range(0,X.shape[0]):
   Z = (Y_pred[i]-y_plus-Xi)/Y_variance[i]
   result[i]= (Y_pred[i]-y_plus-Xi)*stats.norm.cdf(Z) + Y_variance[i]*stats.norm.pdf(Z)
return result
```

3. Random Forests Classifier for MNIST

2. (a) 0.876428790843
   (b) prameters = np.array([5,10,10,5,200])

3. (a)
```
                'min_samples_split': int(params[1]),
                'max_depth': int(params[2]),
                'min_samples_leaf': int(params[3]),
                'max_features': int(params[4])}
```
(b) Sometimes
  0.896714220283
  prameters: 5,18,6,68,166
(c)
```
    objective = black_box.RandomForestObjective(
        X_train=np.load("data/mnistFeatures.npy"),
        y_train=np.load("data/mnistlabels.npy"))
    xv = np.random.randint(1,200,(200,4))
    xu = np.random.randint(1,10,(200,1))
    x = np.vstack((xu.T,xv.T))
    x = x.T
    bo = BO(objective, noise=1e-1)

    for _ in xrange(50):
        bo.optimize(num_iters=1)
```