h092540 Machine Learning HW5

1.Guassian Process

<u>Part1.</u>

use rational quadratic kernel:

$$k(x_a, x_b) = \sigma^2 \left(1 + \frac{\|x_a - x_b\|^2}{2\alpha\ell^2}\right)^{-\alpha}$$

hyper-parameter:

σ : 0.2 (because noisy of data is β = 0.2)

α : 0.2

l : 0.5

code:

kernel:

```
def kernel(self, x1, x2):
    m, n = x1.shape[0],x2.shape[0]
    dist_matrix = np.zeros((m,n),dtype = float)
    dist_matrix_A = []
    for i in range(m):
        for j in range(n):
            dist_matrix[i][j] = np.sum((x1[i]-x2[j])**2)
    #dist_matrix_A = np.sum(x1**2, 1).reshape(-1, 1) + np.sum(x2**2, 1) - 2 * np.dot(x1, x2.T)
    return self.params["sigma_f"] ** 2 * (1 + dist_matrix / (2*self.params["alpha_"]*self.params["l"]))**(-self.params["alpha_"])
```

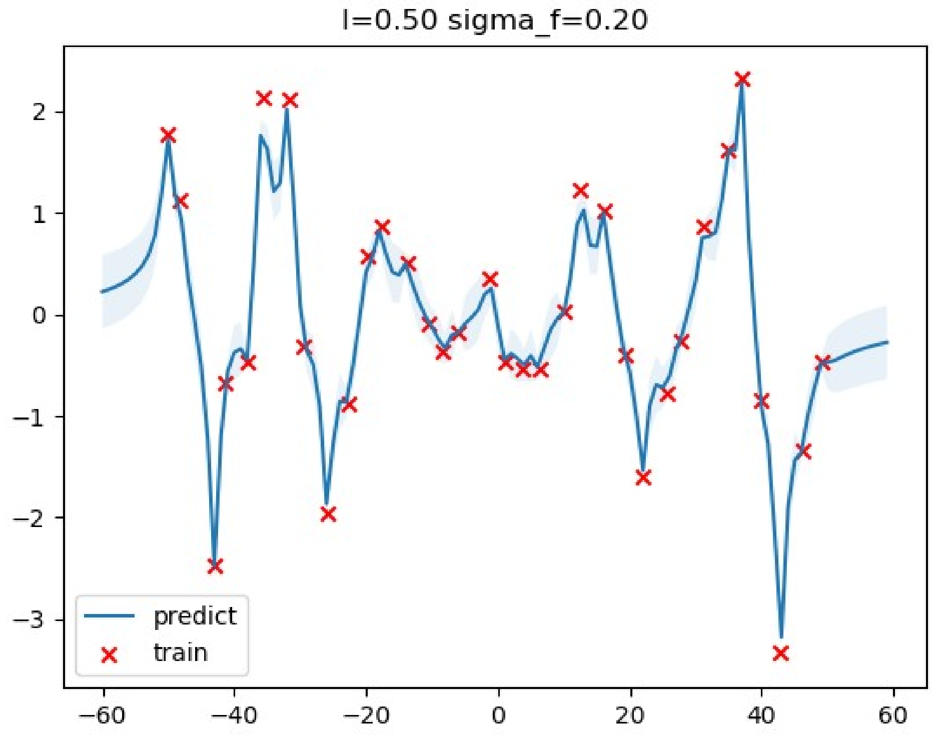
predict(get mean and covariance matrix):

```
def predict(self, X):
    if not self.is_fit:
        print("GPR Model not fit yet.")
        return

    X = np.asarray(X)
    Kff = self.kernel(self.train_X, self.train_X)  # (N, N)
    Kyy = self.kernel(X, X)  # (k, k)
    Kfy = self.kernel(self.train_X, X)  # (N, k)
    Kff_inv = np.linalg.inv(Kff + 1e-8 * np.eye(len(self.train_X)))  # (N, N)

    mu = Kfy.T.dot(Kff_inv).dot(self.train_y)
    cov = Kyy - Kfy.T.dot(Kff_inv).dot(Kfy)
    return mu, cov
```

draw plot:

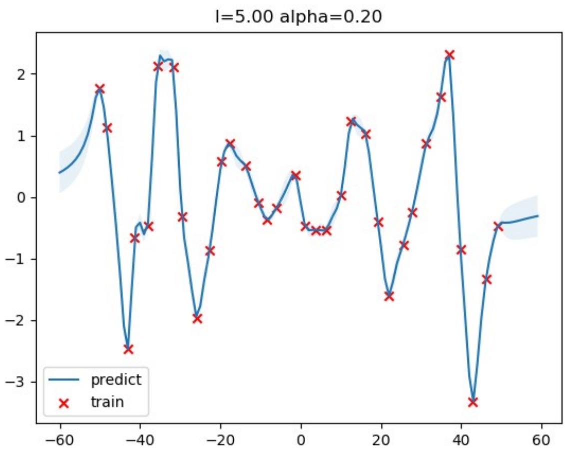
```python
gpr = GPR()
gpr.fit(train_X, train_y)
mu, cov = gpr.predict(test_X)
test_y = mu.ravel()
uncertainty = 1.96 * np.sqrt(np.diag(cov))#95% confidence interval of f
plt.figure()
plt.title("l=%.2f sigma_f=%.2f" % (gpr.params["l"], gpr.sigma_))
plt.fill_between(test_X.ravel(), test_y + uncertainty, test_y - uncertainty, alpha=0.1)
plt.plot(test_X, test_y, label="predict")
plt.scatter(train_X, train_y, label="train", c="red", marker="x")
plt.legend()
plt.show()
```

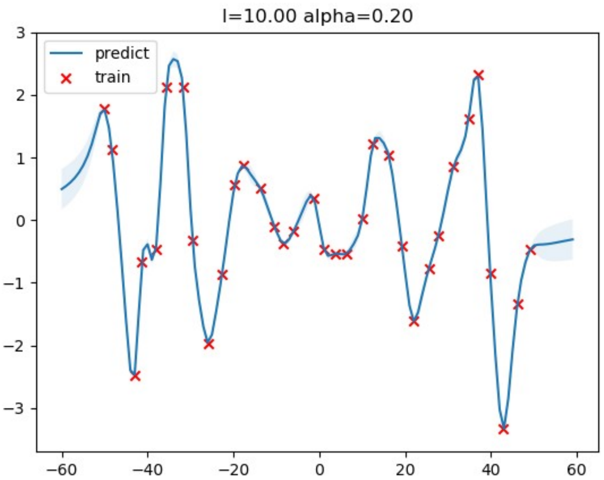
plot:



<u>Part2.</u>
Optimize:

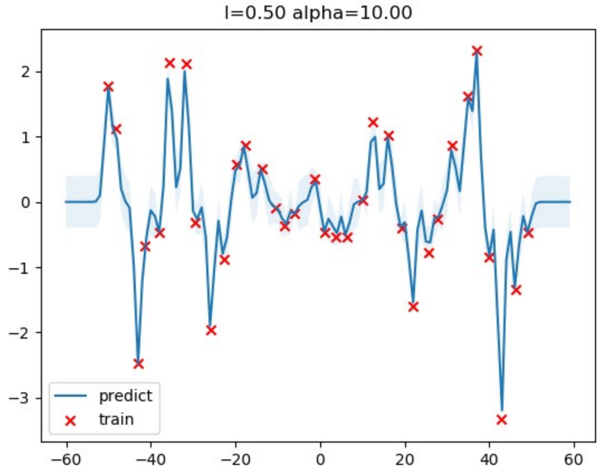
<u>Discussion.</u>
for l:

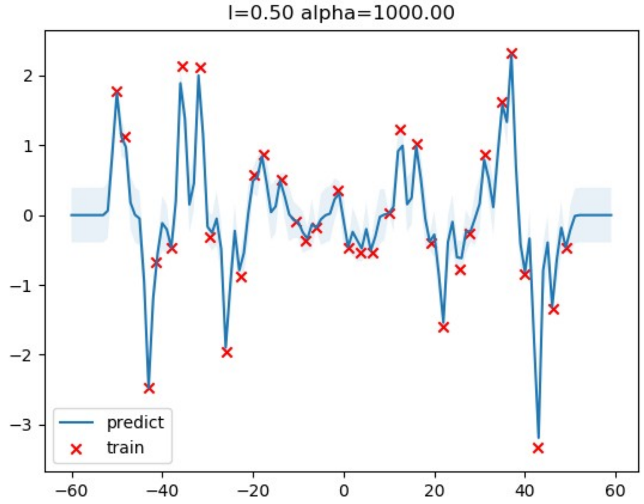




By increasing value of l, the overall spread of the covariance increases . The curve is more smooth.

for alpha:





Decreasing the scale-mixture α will allow for more minor local variations while still keeping the longer scale trends defined by l.

2.SVM on MNIST dataset