bv

April 10, 2019

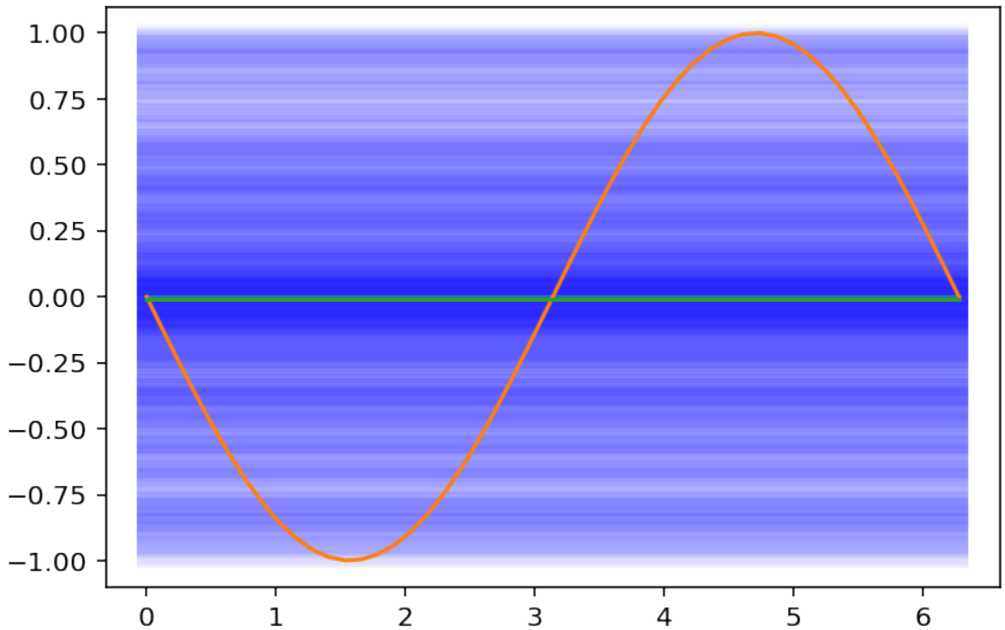
bias(x) = $(\bar{g}(\bar{x}) - f(\bar{x}))^2$
bias = $\mathbb{E}_{\bar{x}}[(\bar{g}(\bar{x}) - f(\bar{x}))^2]$
var(x) = $\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\bar{x}) - \bar{g}(\bar{x}))^2]$
var = $\mathbb{E}_{\bar{x}}[\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(\bar{x}) - \bar{g}(\bar{x}))^2]]$
performance = bias+variance

In [84]:
```
import numpy as np
import matplotlib.pyplot as plt

K = 1000 # how many times we'll chose a random dataset

domain = np.linspace(0,2*np.pi)


B = []
for k in range(K):
    x1 = np.random.rand(1,2)*np.pi*2
    b = np.mean(np.sin(x1))
    plt.plot(domain,b*np.ones(len(domain)),alpha=0.02,c='b',linewidth=7)
    B.append(b)

gbar = np.mean(B)
xx = np.random.rand(1,K)*np.pi*2

A = []
for k in range(K):
    x1 = np.random.rand(1,2)*np.pi*2
    b = np.mean(-np.sin(x1))
    A.append((b-gbar)**2)

var_x = np.mean(A)

var = var_x
bias = np.mean((gbar--np.sin(xx)**2))

print("bias=", bias)
print("variance = ",var)
#plt.scatter(x1,[0,0])
```

```
#plt.scatter(x1,np.sin(x1))
plt.plot(domain,np.zeros(len(domain)))
plt.plot(domain,-np.sin(domain))
plt.plot(domain,gbar*np.ones(len(domain)))
plt.show()
```

```
bias= 0.4927059752060216
variance =  0.2513193143198875
```

Out[84]:



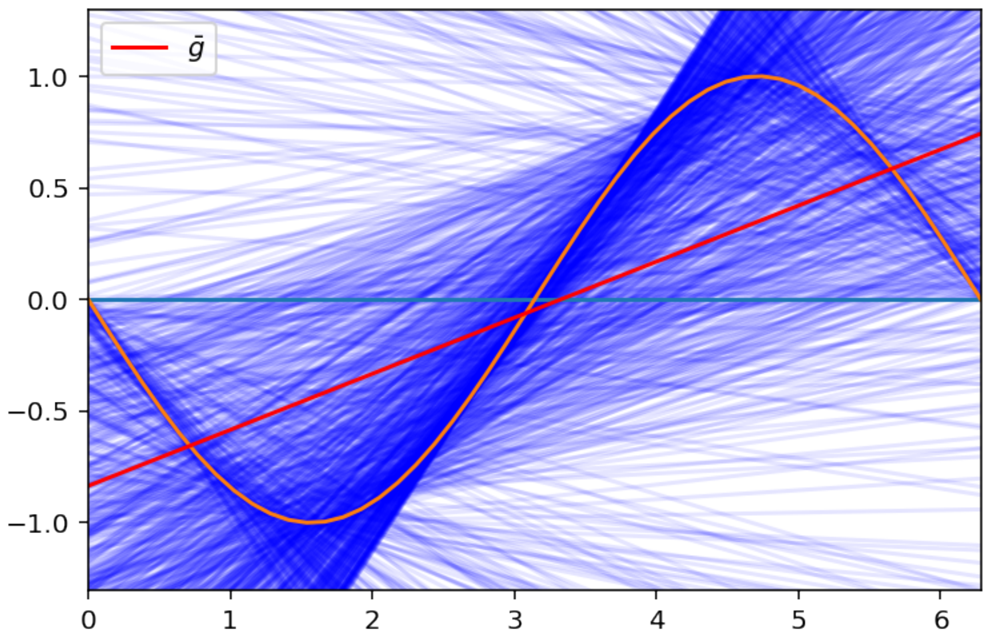
In [83]:

```
import numpy as np
import matplotlib.pyplot as plt

K = 1000 # how many times we'll chose a random dataset

domain = np.linspace(0,2*np.pi)
D = np.c_[np.ones(len(domain)),domain]
G = []
gbar = np.zeros(len(domain)).reshape((len(domain),1))
bunchaxs = np.linspace(0,2*np.pi)
for k in range(K):
    x1 = np.random.rand(1,2)*np.pi*2
    X = np.c_[np.ones(2),x1.T]
    w = np.linalg.pinv(X).dot(-np.sin(x1.T))
```

```python
        plt.plot(domain,D.dot(w),c='b',alpha=0.1)
        gbar += D.dot(w)
        G.append(D.dot(w))
    gbar = gbar/K

    tot = np.zeros(len(domain)).reshape((len(domain),1))
    for g in G:
        tot += (g-gbar)**2

    var_x = tot/len(G)

    var = np.mean(var_x)

    xx = np.random.rand(1,K)*np.pi*2



    bias = np.mean((gbar.ravel()--np.sin(domain))**2)

    print("bias = ",bias)
    print("variance = ",var)
    #plt.scatter(x1,[0,0])
    #plt.scatter(x1,np.sin(x1))
    plt.plot(domain,np.zeros(len(domain)))
    plt.plot(domain,-np.sin(domain))
    plt.plot(domain,gbar,c='r',label=r'$\bar{g}$')
    plt.axis([0,2*np.pi,-1.3,1.3])
    plt.legend()
    plt.show()
```

```
bias =  0.21655930066434248
variance =  1.6671917086482402
```

Out[83]:



In [0]:

In [0]: