GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

**ECE 6250 Spring 2017**
**Problem Set #3**

Assigned: 30-Jan-17
Due Date: 8-Feb-17

---

***Quiz #1 will be held in class on Wednesday, February 15.***

**This assignment is due at the beginning of class.**

**As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.**

---

**PROBLEM 3.1:**
Using you class notes, prepare a 1-2 paragraph summary of what we talked about in class in the last week. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other things you have learned here or in other classes?). The more insight you give, the better.

**PROBLEM 3.2:**
Let $G_1$, $G_2$, and $G_3$ be zero-mean Gaussian random variables with covariance matrix $\boldsymbol{R}$:

$$R_{i,j} = \mathrm{E}[G_i G_j].$$

Define $\mathcal{S} = \mathrm{span}\{G_1, G_2, G_3\}$. That is, $\mathcal{S}$ contains all the random variables $X$ that can be written as $X = a_1G_1 + a_2G_2 + a_3G_3$ for some $a_1, a_2, a_3 \in \mathbb{R}$. It should be clear that all the elements of $\mathcal{S}$ are also zero mean Gaussian random variables.

1. Show that $\langle X, Y\rangle = \mathrm{E}[XY]$ is a valid inner product on the vector space $\mathcal{S}$. Defend the terminology "root mean-square error" (RMSE) for the distance induced by this inner product.

2. Suppose $X = a_1G_1 + a_2G_2 + a_3G_3$ and $Y = b_1G_1 + b_2G_2 + b_3G_3$. Show that $\langle X, Y\rangle = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{b}$, where $\boldsymbol{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^{\mathrm{T}}$.

**PROBLEM 3.3:**
Find $a, b, c \in \mathbb{R}$ that minimize

$$\int_{-1}^{1} |t^3 - a - bt - ct^2|^2 \, dt.$$



Last updated 12:49, January 26, 2017

**PROBLEM 3.4:**

Let the vector $\mathcal{S}$ contain Gaussian random variables as defined in Problem 3.2 with

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0.4 & -0.2 \\ 0.4 & 1 & 0.2 \\ -0.2 & 0.2 & 1 \end{bmatrix}. \tag{1}$$

1. Let $X = G_1$, $Y = G_2$, $Z = G_1 + G_2 + G_3$. Now suppose we observe particular values for $X$ and $Y$, say $X = x$ and $Y = y$. As all three random variables are related to one another, these observations give us some information about the value of $Z$. Here we will consider *linear predictors*: estimates of $Z$ that are linear combinations of the observations; such estimates have the form

   $$\hat{Z} = \alpha_1 X + \alpha_2 Y \quad \alpha_1, \alpha_2 \in \mathbb{R}.$$

   Find the best linear predictor of $Z$. That is, find $\alpha_1, \alpha_2$ so that the mean-square error $\mathrm{E}[(Z - \hat{Z})^2]$ is minimized. Also calculate the actual value of the mean-square error for the best $\alpha_1, \alpha_2$.

   You will want to set this up as an "approximation in a subspace" problem. You also might want to use MATLAB to do some of the calculations.

2. Now suppose $X = a_1G_1 + a_2G_2 + a_3G_3$, $Y = b_1G_1 + b_2G_2 + b_3G_3$, and $Z = c_1G_1 + c_2G_2 + c_3G_3$. Write and MATLAB script that takes $\boldsymbol{R}$, $\boldsymbol{a}, \boldsymbol{b}$, and $\boldsymbol{c}$ as arguments and returns the values of $\alpha_1$ and $\alpha_2$ that minimize $\mathrm{E}[(Z - \hat{Z})^2]$ and the value of the mean-square error for these $\alpha_i$. Turn in a copy of your code.

3. Try your function out on

   $$X = G_1 + 2G_2 + G_3/6, \quad Y = G_1/4 + 5G_2/2 + 2G_3, \quad \text{and} \quad Z = G_1 + G_2 + G_3,$$

   and the covariance matrix $R$ in (1). The file `problem4.mat` contains three arrays `X, Y, Z` that consist of 1000 realizations of each of these random variables. Form `Zhat = alpha1*X + alpha2*Y;` and compute the sample MSE using `mean((Zhat-Z).^2)`. How does it compare to the value your function returned? Finally, does the MSE compare favorably with the variance of $Z$?

**PROBLEM 3.5:**
In this problem, we will develop the computational framework for approximating a continuous-time signal on $[0, 1]$ using scaled and shifted version of the classic bell-curve bump:

$$\phi(t) = e^{-t^2}.$$

Fix an integer $N > 0$ and define $\phi_k(t)$ as

$$\phi_k(t) = \phi\left(\frac{t - (k - 1/2)/N}{1/N}\right) = \phi\,(Nt - k + 1/2)$$

for $k = 1, 2, \ldots, N$. The $\{\phi_k(t)\}$ are a basis for the subspace

$$T_N = \operatorname{span}\left\{\phi_k(t)\right\}_{k=1}^{N}.$$

1. For a fixed value of $N$, we can plot all of the $\phi_k(t)$ on the same set of axes in MATLAB using:

   ```
   phi = @(z) exp(-z.^2);
   t = linspace(0, 1, 1000);
   figure(1); clf
   hold on
   for kk = 1:N
      plot(t, phi(N*t - kk + 1/2))
   end
   ```

   Do this for $N = 10$ and $N = 25$ and turn in your plots.

2. Since $\{\phi_k(t)\}$ is a basis for $T_N$, we can write any $y(t) \in T_N$ as

   $$y(t) = \sum_{k=1}^{N} a_k \phi_k(t)$$

   for some set of coefficients $a_1, \ldots, a_N \in \mathbb{R}^N$. If these coefficients are stacked in an $N$-vector `a` in MATLAB, we can plot $y(t)$ using

   ```
   t = linspace(0,1,1000);
   y = zeros(size(t));
   for jj = 1:N
      y = y + a(jj)*phi(N*t - jj + 1/2);
   end
   plot(t, y)
   ```

   Do this for $N = 4$, and $a_1 = 1, a_2 = -1, a_3 = 1, a_4 = -1$ and turn in your plot.

3. Define the continuous-time signal $x(t)$ on $[0, 1]$ as

   $$x(t) = \begin{cases} 0.5 & 0 \le t < 1/2 \\ -\sin(2\pi t) & 1/2 \le t \le 1 \end{cases}.$$

Write MATLAB code that finds the closest point $\hat{x}(t)$ in $T_N$ to $x(t)$ for any fixed $N$. By "closest point", we mean that $\hat{x}(t)$ is the solution to

$$\min_{y \in T_N} \|x(t) - y(t)\|_{L_2([0,1])}.$$

Turn in your code and two plots; one of which has $x(t)$ and $\hat{x}(t)$ plotted on the same set of axes for $N = 10$, and the other which has the same for $N = 25$.
**Hint:** You can create a function pointer for $x(t)$ using

```
x = @(z) (z < 1/2).*1/2 - (z>=1/2).*sin(2*pi*z);
```

and then calculate the continuous-time inner product $\langle x, \phi_k \rangle$ with

```
x_phik = @(z) x(z).*phi(N*z - jj + 1/2);
quad(x_phik, 0, 1)
```

You can use similar code to calculate the entries of the Gram matrix $\langle \phi_j, \phi_k \rangle$. (There is actually a not-that-hard way to calculate the $\langle \phi_j, \phi_k \rangle$ analytically that you can derive if you are feeling industrious — just think about what happens when you convolve a bump with itself.)