Assignment 8

1. General Comments.

   - You write (7.1): "BB^T is invertible." Look at B being an $n \times 1$ matrix (just a vector $\mathbf{b}$). If $\mathbf{v}$ is a non zero vector perpendicular to $\mathbf{b}$ (i.e. $\mathbf{b}^T\mathbf{v} = 0$, then $BB^T\mathbf{v} = \mathbf{b}\mathbf{b}^T\mathbf{v} = 0$. The matrix BB^T does not have to be invertible. However if $BB^T\mathbf{v}_i = \lambda_i\mathbf{v}_i$, then $0 \le \left|B^T\mathbf{v}_i\right|^2 = \mathbf{v}_i^T BB^T\mathbf{v}_i = \lambda_i\mathbf{v}_i^T\mathbf{v}_i = \lambda_i$.
   - (7.2) is fine.
   - (7.3) is fine.
   - (7.4) is fine.
   - (7.5) a) The solution is fine, but you are working too hard. Indeed, if

     $$\mathbf{w} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_n\mathbf{v}_n,$$

     then

     $$M\mathbf{w} = M\left(c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots + c_n\mathbf{v}_n\right) = c_1M\mathbf{v}_1 + \ldots + c_nM\mathbf{v}_n = c_1\lambda_1\mathbf{v}_1 + \ldots + c_n\lambda_n\mathbf{v}_n.$$

   - (7.5) b) and c) are fine
   - as far as (7.5) d) is concerned I have a comment and a question:
     (a) comment. You write "Let λ_j be the maximum eigenvalue. Multiply by $\frac{1}{\lambda_j^{2k}c^2}$." I guess you mean $\frac{1}{\lambda_j^{2k}c_j^2}$.
     (b) question. What do you suggest to do if $c_j = 0$?

2. The method described by Problem 7.5 (grab a vector, multiply this vector by the matrix M and normalize the result. The sequence of vectors generated this way converges to the eigenvector corresponding to the largest eigenvalue of the matrix M) is called the power method. This is the simplest numerical method to determine the largest eigenvalue and the corresponding eigenvector of a symmetric matrix with non negative eigenvalues.

   The next problem is coding.

   Problem 8.1. For a given set of vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ write a code that from the input outlined below generates output described below.

   Input:

   m-the number of vectors

n-dimension

File with $n \times m$ lines, each line contains a double (a single coordinate of a vector).

Output:$\mathbf{v}$ the eigenvector corresponding to the largest eigenvalue of the matrix $\left(A - \mathbf{c}\mathbf{e}^T\right)\left(A - \mathbf{c}\mathbf{e}^T\right)^T$, where $\mathbf{c} = \dfrac{\mathbf{a}_1 + \ldots + \mathbf{a}_m}{m}$, and $\mathbf{e}$ is an n dimensional vector with entries 1, that is $\mathbf{e} = \begin{bmatrix} 1 \\ \ldots \\ 1 \end{bmatrix}$.

I note the following:

(a) This code is a module in a code needed for implementation of PDDP. At the first iteration PDDP uses the entire dataset $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$. At the second and following steps PDDP uses one of the clusters generated at the previous step, i.e. only a part of the dataset $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ is used to build the matrix B. The code in Problem 8.1 should be able to use a predefined part of the dataset.

(b) The power method that will be implemented multiplies a vector $\mathbf{w}$ by the matrix $\left(A - \mathbf{c}\mathbf{e}^T\right)\left(A - \mathbf{c}\mathbf{e}^T\right)^T$. Note that $\left(A - \mathbf{c}\mathbf{e}^T\right)^T \mathbf{w} = A^T\mathbf{w} - \mathbf{e}\left(\mathbf{c}^T\mathbf{w}\right)$. This suggests that computation of $A - \mathbf{c}\mathbf{e}^T$ is not necessary (hence some memory and time can be saved).