**Probabilistic PCA**

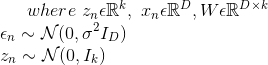
In this lab, you will implement probabilistic version of PCA. Basically, it is all about implementing Expectation-Maximization algorithm. First we use EM-algorithm to get transformation matrix from high dimensional data space to low-dimensional space. Then we use Gaussian classifier for classification on this low-dimensional representation.

Probabilistic PCA (pPCA) model and EM-algorithm are given as follows:

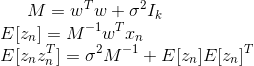
**Probabilistic PCA model:**

Assuming data is centered, we define the model as follows, x\_n is a column data vector,





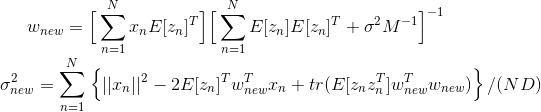
**E-step:**



**Inverse of M can be efficiently calculated in the following way:**



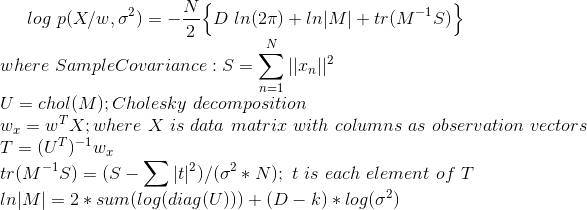
**M-step:**



**Now update old parameters with new parameters**



**Calculating log-likelihood:**



All these equations can be found in 12th chapter of Pattern Recognition and Machine Learning textbook except for explicit details of computation of likelihood.

**Data:** Use MNIST dataset you have used in previous labs.

Procedure:

1. Initialize W and sigma with random values. Experiment with k for different values 50, 100. Remember to center the data before doing anything.
2. Calculate log-likelihood
3. Using E-step obtain mean and variance of hidden variables z\_n
4. Using M-step, obtain new values for W and sigma and update old parameters with these new parameters
5. Repeat steps 2,3,4 until log-likelihood converges. Make sure log-likelihood is increasing in each iteration
6. Now you have obtained w, transformation matrix, using which you can obtain low-dimensional representation of the data
7. Project the data onto pPCA space using w
8. Now, Classify using Gaussian classifier with individual covariance matrices.