**Q1.**  
  
In the last week of the course, we learnt about how to fit models to data. These belong to 'unsupervised learning'. The models can be geometric or probabilistic. We explored, using PCA, how to fit a subspace to a bunch of points. PCA is a technique to learn the subspace from the data given, by basically solving an optimization problem. PCA is an example of Structured Matrix Factorization. We discussed the concepts behind Eigen faces and Latent Semantic Indexing. We then discussed about Non-negative Matrix Factorization to use only positive factors. In signal processing, it is possible to take a piece of data and transform it in such a way that it is sparse. Sparse transforms gives us a natural way to compress signals and to remove noise. Sparse regression refers to a general class of techniques to solve systems of equations using as few variables as possible. We also learnt about Dictionary Learning, as another example of structured matrix factorization. In the last class, we talked about mixture models, which simply means that the pdf for a random variable we are modelling is a weighted sum of different component pdfs. This concept is used in demodulation of digital signals, pixel clustering and binary images. A particular mixture model is the Gaussian Mixture Model. GMM for a random variable means its pdf is a convex combination of normal pdfs with different means and different covariance matrices. Data points is represented in each ellipsoid, in different classes. We also talked about estimating the parameters of GMM using the Expectation-Maximization (EM) Algorithm.

**Q2.**

CIOS course evaluation survey done.

**Q3.  
Code:**

syms x1 x2

f1(x1,x2) = (1/(2\*pi\*sqrt(18)))\*(exp(-(1/36)\*(18\*x1^2 + 3\*x2^2 + 12\*x1\*x2 - 12\*x1 - 6\*x2 + 3)));

f2(x1,x2) = (1/(2\*pi))\*(exp(-0.5\*(x1^2 + x2^2 - 2\*x1 + 1)));

f3(x1,x2) = (-5)\*x2^2 + (4)\*x1\*x2 - (8)\*x1 - (2)\*x2 - (5) - 12\*log(1/sqrt(18));

ezsurf(f3);

hold on;

% eqn = (-5)\*x2^2 + (4)\*x1\*x2 - (8)\*x1 - (2)\*x2 - (5) - 12\*log(1/sqrt(18));

% sol = solve(eqn);

y2 = linspace(-6,6,1000);

for i = 1:1000

y1(i) = (5\*y2(i)^2 + 2\*y2(i) - 868507263970121/70368744177664)/(4\*y2(i) - 8);

end

figure;

plot(y1,y2,'ko');

%axis([-6 6 -6 6])

% Now evaluate random points for f1,f2 and f3 and see which is greater.









