10-715 Advanced Introduction to Machine Learning

Homework 3 *Due Nov 12, 10.30 am*

Rules

1. Homework is due on the due date at 10.30 am. Please hand over your homework at the beginning of class. *Please see course website for policy on late submission.*
2. We recommend that you typeset your homework using appropriate software such as LaTeX . If you are writing please make sure your homework is cleanly written up and legible. The TAs will not invest undue effort to decrypt bad handwriting.
3. You must hand in a hard copy of the homework. The only exception is if you are out of town in which case you can email your homeworks to *both* the TAs. If this is the case, your homeworks *must* be typeset using proper software. Please do *not* email written and scanned copies. Your email must reach the TAs by 10.30 am on the due date.
4. You are allowed to collaborate on the homework, but should write up your own solution and code. Please indicate your collaborators in your submission.
5. Please hand in the solutions to Problems 1,2 and Problem 3 separately. Write your name, andrew id and department on both submissions.
6. Please staple your homeworks.
7. If you are confused about of any of the terms you may refer Wikipedia. We have introduced some new concepts and methods that were not discussed in class. You should be able to find all of the required definitions on Wikipedia.

---

1 Dimensionality Reduction (Samy)

1.1 Principal Components Analysis

Principal Components Analysis (PCA) is a popular method for linear dimensionality reduction.

PCA attempts to find a lower dimensional subspace such that when you project the data onto the subspace as much of the variance is preserved. Say we have data $X = [x_1^\top; \ldots; x_n^\top] \in \mathbb{R}^{n \times D}$ where each point is $x = [x^{(1)}, \ldots, x^{(D)}] \in \mathbb{R}^D$. We wish to find a d $(< D)$ dimensional subspace $A = [a_1, \ldots, a_d] \in \mathbb{R}^{D \times d}$ $a_i \in \mathbb{R}^D$, $A^\top A = I_d$, so as to maximize $\frac{1}{n}\sum_{i=1}^n \|A^\top x_i\|^2$.

1. **(2 Points)** We want to find a_1 to maximize $\frac{1}{n}\sum_i (a_1^\top X_i)^2$. Show that a_1 is the first right singular vector of X.
2. **(3 Points)** Given $a_1, \ldots, a_k$, let $A_k = [a_1, \ldots, a_k]$ and $\tilde{x}_i = x_i - AA^\top x_i$. We wish to find a_{k+1}, to maximize $\frac{1}{n}\sum_i (a_{k+1}^\top \tilde{x}_i)^2$. Show that a_{k+1} is the $(k+1)^{th}$ right singular vector of X.

1.2 Affine Subspace Identification (ASI)

We will now motivate the dimensionality reduction problem in a slightly different perspective. The resulting algorithm has many similarities to PCA. We will refer to method as Affice Subspace Identification (ASI).

You are given data $(x_i)_{i=1}^n$, $x_i \in \mathbb{R}^D$. Let $X = [x_1^\top; \ldots x_n^\top] \in \mathbb{R}^{n \times D}$. We suspect that the data actually lies in a d dimensional affine subspace plus some gaussian noise. Our objective is to find a d dimensional representation z for x–this can be used as a preprocessing step before an algorithm. In particular, we are *not* after any interpretation for this lower dimensional representation.

We will assume $d < n$ and that the span of the data has dimension larger than d. Further, our method should work whether $n > D$ or $n < D$.

We wish to find parameters $A \in \mathbb{R}^{D \times d}$, $b \in \mathbb{R}^D$ and a lower dimensional representation $Z \in \mathbb{R}^{n \times d}$ so as to minimize

$$J(A, b, Z) = \frac{1}{n}\sum_{i=1}^n \|x_i - Az_i - b\|^2.$$

Here $Z = [z_1^\top; \ldots; z_n^\top]$ and z_i is the representation for x_i.

1. **(1 Point)** Let $C \in \mathbb{R}^{d \times d}$ be invertible and $d \in \mathbb{R}^d$. Show that both (A_1, b_1, Z_1) and (A_2, b_2, Z_2) achieve the same value on the objective J. Here, $A_2 = A_1 C^{-1}$, $b_2 = b_1 - A_1 C^{-1} d$ and $Z_2 = Z_1 C^\top + \mathbf{1} d^\top$.

Therefore in order to make the problem determined we need to impose some constraint on Z. We will assume that the z_i's have zero mean covariance Ψ where Ψ is given.

$$\bar{Z} = \frac{1}{n}\sum_{i=1}^n z_i = 0, \qquad S = \frac{1}{n}\sum_{i=1}^n z_i z_i^\top = \Psi$$

1. **(6 Points)** Outline a procedure to solve the above problem. Specify how you would obtain A, Z, b which minimize the objective and satisfy the constraints.
   Hint: The rank k approximation of a matrix in Frobenius norm is obtained by taking its SVD and then zeroing out all but the first k singular values.
2. **(1 Point)** You are given a new point x_*. Give the rule to obtain the d dimensional representation x_* for the new point.

1.3 Factor Analysis (FA)

Factor analysis is a generative model for linear dimensionality reduction. As before we will assume a $d(< n)$ dimensional latent space. However, this time we assume the following generative process for the data.

$$\mathbb{R}^d \ni z \sim \mathcal{N}(\mathbf{0}, \Psi)$$
$$\mathbb{R}^D \ni x|z \sim \mathcal{N}(Az + b, \eta^2 I)$$

where Ψ is already known. The model says that we first sample a d dimensional Gaussian with zero mean and variance Ψ. Then we map it to D dimensions by computing $Az + b$. Finally, we add some spherical Gaussian noise with variance η^2 on each dimension.

We will use an EM procedure to learn the parameters A, b, η. So far we were only looking at EM with discrete latent variables. In this case we will look at EM with a parametric continuous latent space.

The following results will be useful for us:

Fact 1 (Conditional of a Gaussian)**.** *Say $(Y_1, Y_2), Y_i \in \mathbb{R}^{d_i}$ is Gaussian distribued.*

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22} \end{bmatrix}\right)$$

Then, conditioned on $Y_1 = y_1$ the distribution for Y_2 is

$$Y_2|Y_1 = y_1 \sim \mathcal{N}(\mu_2 + \Sigma_{12}^\top \Sigma_{11}^{-1}(y_1 - \mu_1),\ \Sigma_{22} - \Sigma_{12}^\top \Sigma_{11}^{-1} \Sigma_{12})$$

Fact 2 (Some Matrix Derivatives)**.** *Let $X \in \mathbb{R}^{r \times c}$, and $u \in \mathbb{R}^r$, $v, w \in \mathbb{R}^c$.*

$$\nabla_X v^\top X^\top u = uv^\top$$
$$\nabla_X v^\top X^\top X w = X(vw^\top + wv^\top)$$

1. **(4 Points)** Write down the joint distribution of (z, x). Use this to derive the marginal distribution of x and the conditional distribution $z|x$.
   The conditional distribution $z|x$ will be useful for us when performing the E-step. In addition, one option for the lower dimensional representation of x_i will be the conditional mean $\mathbb{E}[z|x_i]$.
2. **(3 Points)** First obtain the Maximum Likelihood Estimate for b. This does not require EM and can be done easily.
3. **(1 Point)** Obtain a lower bound on the log likelihood using $R(z_i|x_i)$ - the conditional distribution for z_i given x_i.
4. **(2 Points)** Write down the E-step update at the $(t+1)^{th}$ iteration. Here, you compute $R(z_i|x_i)$ for all data points using your estimates at the t^{th} iteration.
5. **(5 Points)** Now write down the M-step. You need to maximize the lower bound w.r.t A and η.

1.4 Experiment

Here we will compare the above three methods on two data sets. For ASI and FA we will take $\Psi = I_d$.

A common preprocessing step before applying PCA is to subtract the mean. As we will see, without this preprocessing just taking the SVD of X will give very bad results. For the purposes of this problem we will call these two variants “demeaned PCA” and “buggy PCA”. Sometimes, after subtracting the mean we also apply a diagonal scaling so that each dimension has unit variance. We will call this normalized PCA.

One way to study how well the low dimensional representation captures the linear structure in our data is to project it back to D dimensions and look at the reconstruction error. For PCA, if we mapped it to d dimensions via $z = Vx$ then the reconstruction is $V^\top z$. For the preprocessed versions, we first do this and then reverse the preprocessing steps too. For ASI we just compute $Az + b$. For FA, we will use the posterior mean $\mathbb{E}[z|x]$ as the lower dimensional representation and $Az + b$ as the reconstruction. We will compare all four methods by the reconstruction error on the datasets.

Please implement code for the five methods: Buggy PCA (just take the SVD of X) , Demeaned PCA, Normalized PCA, ASI, FA. In all cases your function should take in an $n \times d$ data matrix and d as an argument. It should return the the d dimensional representations, the estimated parameters, and the reconstructions of these representations in D dimensions. For FA, use the values obtained from ASI as initializations for A. Set η based on the reconstruction errors of ASI. Use 10 iterations of EM.

You are given two datasets: A two Dimensional dataset with 50 points `data2D.mat` and a thousand dimensional dataset with 500 points `data1000D.mat`.

For the $2D$ dataset use $d = 1$. For the $1000D$ dataset, you need to choose d. For this, observe the singular values in ASI and see if there is a clear "knee point" in the spectrum. Attach any figures/ Statistics you computed for this to justify your choice.

For the $2D$ dataset you need to attach the a plot comparing the orignal points with the reconstructed points for all five methods. For both datasets you should also report the reconstruction errors. The given starter code does all of this for you so you need to just attach the results.

These were our errors for the $2D$ dataset. If your answers do not tally, please check with the TAs.

```
>> q14
Reconstruction Errors:
Buggy PCA: 0.365284
Demeaned PCA: 0.008448
Normalized PCA: 0.008454
ASI: 0.008448
FA: 0.008526
```

Questions

1. Look at the results for Buggy PCA. The reconstruction error is bad and the reconstructed points don't seem to well represent the original points. Why is this ?
   Hint: Which subspace is Buggy PCA trying to project the points onto ?

2. In both demeaned PCA and ASI the errors are identical. But the Z values are not. You can check this via the command `norm(Z2-Z4, 'fro')`. Explain why ?.

3. The error criterion we are using is the average squared error between the original points and the reconstructed points. In both examples ASI (and demeaned PCA) achieves the lowest error among all methods. Is this surprising ? Why ?

Please submit your code along with your results.

Point Allocation

- Implementation of all five methods: **(6 Points)**
- Results - Figures and errors **(3 Points)**
- Choice of d for $1000D$ dataset and appropriate justification: **(1 Point)**
- Questions **(3 Points)**

2 Some Random Topics (Samy)

2.1 K-means Clustering

1. **(3 Points)** Given n observations $X_1^n = X_1, \ldots, X_n$, $X_i \in \mathcal{X}$ the K-means objective is to find k $(< n)$ centres $\mu_1^k = \mu_1, \ldots, \mu_k$ and a rule $f : \mathcal{X} \to \{1, \ldots, K\}$ so as to minimize the objective

$$J(\mu_1^K, f; X_1^n) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}(f(X_i) = k) \|X_i - \mu_k\|^2$$

   Let $\mathcal{J}_K(X_1^n) = \min_{\mu_1^k, f} J(\mu_1^K, f; X_1^n)$. Prove that $\mathcal{J}_K(X_1^n)$ is a non-increasing function of K.

2. **(2 Points)** Consider the K-means clustering algorithm we studied in class. We terminate the algorithm when there are no changes to the objective. Show that the algorithm terminates in a finite number of steps.

2.2 Independent Components Analysis

1. **(3 Points)** You are given starter code that generates some signals in Matlab and then mixes them. Use any ICA library (FastICA is a popular package, http://research.ics.aalto.fi/ica/fastica/) and reconstruct the signals. Attach your code and the unmixed signals.

2. **(1 Point)** Explain why the unmixed components may be scaled versions of the original inputs.

3 Graphical Models (Veeru)

1. **(3 points)** Consider a situation where there is a student is interested in getting a letter from his course instructor. The instructor writes a letter based on the student's grade which in turn depends on how smart the student is and how much effort he has put into the course. Consider the graphical model in Figure (1) which encodes these dependencies between the variables: Intelligence(I), Work(W), Grade(G), Letter(L).

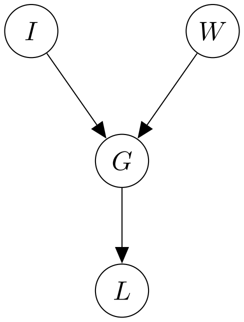

Figure 1: Graphical model for Problem 1

   (a) Write the joint probability distribution in the factored form for the given graphical model.

   (b) Assuming that there are $d = 10$ levels in each of the variables(that is, each variable takes d values with non-zero probability), then how many parameters do we need to model the full joint distribution, with and without the knowledge of dependencies in the graphical model.

2. **(4 points)** Do the following independencies hold? Justify.

(a) $I \perp W|L$

(b) $I \perp G$

(c) $I \perp L|G$

(d) $G \perp L|W$

3. **(4 points)** In this part, you will compute some conditional probabilities based on the graphical model. Assume I, W, L are binary variables taking values from $\{0, 1\}$ and $G \in \{0, 1, 2\}$. The conditional probabilities are given by:

- $P(I = 1) = 0.7$, $P(W = 1) = 0.8$
- $P(L = 1|G = 0) = 0, P(L = 1|G = 1) = 0.3, P(L = 1|G = 2) = 0.8.$

	$P(G\|I = 0, W = 0)$	$P(G\|I = 0, W = 1)$	$P(G\|I = 1, W = 0)$	$P(G\|I = 1, W = 1)$
G=0	0.9	0.1	0.1	0
G=1	0.1	0.6	0.7	0.1
G=2	0.0	0.3	0.2	0.9

Now answer the following questions.

(a) What is the probability that a student gets a letter?

(b) What is the probability that a student gets a letter provided he is intelligent but does not work?

4. **(4+5 points) Hidden Markov Models**

HMMs can be used to infer hidden state sequences from time series data. Suppose there is a soccer striker who, on a given day, plays in one of three modes: bad(0), average(1), good(2). Roughly speaking, he scores more goals as he changes his mode from bad through good. Further, he has a tendency to play in a fixed mode for a few games before he changes the mode. Formally, let x_t denote the mode he is in game t and y_t denote the number of goals he scores. The conditional probability $p(y|x)$ is given by for all t:

	$p(y = 0\|x)$	$p(y = 1\|x)$	$p(y = 2\|x)$	$p(y \geq 3\|x)$
$x = 0$	0.9	0.1	0	0
$x = 1$	0.8	0.1	0.1	0
$x = 2$	0.5	0.3	0.1	0.1

He goes from mode i to mode j between games with probability T_{ij} where

$$T = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$

Suppose he scores the following number of goals in 12 consecutive games:

1 5 0 1 1 0 0 0 0 0 0 0

Assume that for the first game, his mode is uniformly distributed.

(a) What is the probability of this sequence of goals?

(b) What is the most likely sequence of his modes in these games?

If you program, please attach your code. In the unlikely case that you work out by hand, please include all calculations.

4 Markov Chain Monte Carlo(Veeru)

4.1 Markov Chain properties

Let T be the transition probability matrix of a Markov Chain. So T_{ij} is the probability of going from state i to state j, for $i, j \in [n]$ where n is the size of the state space.

1. **(2 points)** Show that 1 is an eigen value of T^T.
2. **(3 points)** Show that the Markov Chain with the following transition matrix does not have a unique stationary distribution:

$$T = \frac{1}{10}\begin{bmatrix} 3 & 0 & 7 & 0 \\ 0 & 4 & 0 & 6 \\ 4 & 0 & 6 & 0 \\ 0 & 7 & 0 & 3 \end{bmatrix}$$

3. **(2 points)** This is a long question, but needs a short answer. Suppose a random walker starts on a Markov Chain with a distribution p_0 over the states and makes transitions according to the transition probability matrix T. The probabilities of the random walker over the states, after n steps are given by $p_0^T T^n$. It is well-known that the largest absolute value of an eigen-value of T is 1. Let E be the space formed by the eigen-vectors of T whose associated eigen-value is 1 in absolute value. If we make sufficiently large number of transitions n, the projection of p_0 in the space perpendicular to E goes to 0 as $n \to \infty$. (To see this, let F be the eigen-space of T^T and write p_0 as the sum of its projections $P_F(p_0), P_{F^\perp}(p_0)$ on F and $F^\perp$ where $F^\perp$ is the space orthogonal to F. $P_F(p_0)$ can be written a linear combination of eigen vectors of T^T and the eigen values of $(T^T)^n$ are the eigen values of T^T raised to n etc.). So after sufficiently large number of transitions, we hope to converge to a stationary distribution.

   Find a T and a p_0 such that $p_0^T T^n$ fails to converge to a stationary distribution of T.

4.2 Detailed balance property

1. **(4 points)** Show that the transition kernel in Metropolis Hastings algorithm satisfies detailed balance property. Assume that the proposal density and the target density are positive.
2. **(2 points)** Show that the detailed balance property is sufficient for the existence of a stationary distribution. Note that it is not necessary for existence.

4.3 Experiments

1. **(3+3+2 points) Metropolis Hastings**

   Pretend that you do not know how to draw samples from a mixture of Gaussians. We wish to draw samples from:

$$p(x) \propto \exp(-\frac{1}{2}(x-\mu)^2) + \exp(-\frac{1}{2}(x+\mu)^2))$$

   with $\mu = 5$ using Metropolis Hastings(MH) algorithm. Let the proposal distribution $q(x'|x)$ be $N(x, \sigma^2)$. We will vary σ and see how MH behaves.

   (a) Implement MH with $\sigma = 0.5$ and initial point $x_0 = 0$. Discard the first $b = 10000$ samples and compute the mean of the next $n = 1000$ samples. Repeat this $m = 6$ times and plot the $b + n$ samples(including the samples collected during the burnin period) with m subplots for the m repititions. What are the sample means(after the burnin period) in the m cases? You should report m numbers. The mean of the distribution p is clearly 0. If there is a discrepancy between the sample means and the population mean, explain.

(b) Repeat the experiment with $\sigma = 5$. What are the m sample means this time? Is this σ better than the earlier one? Explain your findings.

(c) Report the average of the sample means if we take a larger m, say $m = 50$, while still maintaining $\sigma = 0.5$.

2. **(3+6 points) Gibbs sampling for Gaussian Mixture models**
Suppose we have a data points $x_1, x_2, \cdots, x_n \in \mathbb{R}^2$ that we think are generated from a mixture of K Gaussians with unit variance. Let $\mu_1, \mu_2, \cdots, \mu_K$ denote the K cluster centers and let $z_1, z_2, \cdots, z_n$ be the cluster assignments. For brevity, denote $z = \{z_1, \cdots, z_n\}$, $\mu = \{\mu_1, \cdots, \mu_K\}$ and $x = \{x_1, \cdots, x_n\}$. In this part, we would like to treat z, μ as the latent variables and use Gibbs sampling to sample from the posterior $p(z, \mu|x)$.

(a) For Gibbs sampling, we need to find the conditional distributions of the latent variables z_i, μ_k given the rest of the variables. Show that

$$p(z_i = k|x, z_{-i}, \mu) \propto p(x_i|z_i = k, \mu_k)p(z_i = k), \tag{1}$$

$$p(\mu_k = u|x, z, \mu_{-k}) \propto p(\mu_k = u) \prod_{\{i:z_i=k\}} p(x_i|z_i = k, \mu_k = u) \tag{2}$$

We need to assume priors over z, μ to fully specify these conditionals. For simplicity, for $i = 1, 2, \cdots, n$, $k = 1, 2, \cdots, K$, let the priors be

$$p(z_i = k) = \frac{1}{K}, \qquad \mu_k \sim N(0, I).$$

Note: There was an additional $p(\mu_k)$ term in (1) earlier.

(b) Generate 30 points each from three Gaussian distributions centered at $(-4, 0), (4, 0)$ and $(0, 7)$ with unit variance. Let $K = 3$. Using the conditional distributions in (1), (2) and the given priors, run Gibbs sampling starting at random points drawn from the priors, discard the first 5000 samples and for the next sample, scatter plot the points $x_1, x_2, \cdots, x_n$ with a marker/color-coding depending on the cluster assignments $z_1, z_2, \cdots, z_n$. That is, make sure that the cluster assignments are visible when you plot the points.