IML Exercise 1 Answers

1. Theoretical part
   1. Mathematical Background
      1. Linear Algebra
2. For any orthogonal matrix the linear transformation by is isometric. i.e.:

Proof: Orthogonal Matrix is made of columns that are orthonormal vectors, i.e. they are   
and vectors are mutually perpendicular.  
and the And for the reciprocal matrix

~~So as a result, can be represented by this base:~~

For simplicity we’ll prove the square of each norm Since the norms are in even after taking the root we know it is not negative.

So and hence:

1. We’ll calculate SVD for

We can start the decomposition by wither calculating or

Smaller and easier:

So:

Now for computing we will make EVD that we already know it’s e.vals and we need only to compute the e.vecs:

For :

Normalized by:

For :

Normalized by:

For :

Normalized by:

So collecting overall e.vecs:

If we want e.vals sorted in the matrix it will look like

1. We’ll prove the power-iteration algorithm convergence to (e.vec of )  
   when and initial selected has

First we’ll prove that by recursion

For by definition, Assuming :

Now when and is a diagonal matrix with on its diagonal.

Now let’s examine:

So:

\* - Since it dominants when k goes to infinity.

* + 1. Multivariate calculus

1. For fixed, fixed orthogonal matrix,

Since is orthogonal matrix it’s columns are vectors that spans so we can express : or and to get this

So the Jacobian:

We’ll examine on specific index :

1. For we want to find

We got:

Define:

We’ll leaned in the recitation that for : Short proof:

1. The soft-max function is

We’ll assume as was answered in the exercise forum.   
The Q’s is

By

We have

So the Jacobian is

Note it is a symmetric matrix.   
It is positive on the diagonal and negative on the rest of the entries.

1. The function , we’ll find the Hessian of
   1. Estimation Theory
2. with finite.

We look on first and the mean estimator

This estimator is unbiased as we saw in the lecture and   
We’ll show it is also consistent meaning the probability is concentrated around the expected value with

By Chebyshev:

As we saw in the lecture

So for every finite given the bounding is:

1. The sample set: are finite.

We saw in the lecture the PDF of a sample is:

The likelihood of the i.i.d. samples set will be:

So the log-likelihood will be:

While for a single sample:

1. Practical part
   1. Univariate Gaussian Estimation
2. From a 1000 samples of normal distribution of with np.random.seed(0), we got estimated mean , variance (unbiased estimator) of:

(9.954743292509804, 0.9752096659781323)

Calculated by:

1. When sample set size is increasing from 10 to 1000 only on the samples set we already took on Q1, the consistency is demonstrated.

Text

Description automatically generated with low confidence

1. The Probability-Density-Function of the values in the data set of 1000 samples of with np.random.seed(0), is compared here versus the ideal PDF.

We can see we under-estimated the variance (0.975) and the ideal was slightly higher (1.0)

The PDF of the sample points are on the estimated normal distribution model.

Chart, line chart

Description automatically generated

* 1. Multivariate Gaussian Estimation

1. From a 1000 samples of normal distribution of when

with np.random.seed(0),

We estimate expected value by:

We estimate variance value by:

we got estimated mean , variance (unbiased estimator) of:

Estimated mu vector is

[-0.02282878 -0.04313959 3.9932571 -0.02038981]

Estimated Covariance (Sigma) matrix is

[[ 0.91667608 0.16634444 -0.03027563 0.46288271]

[ 0.16634444 1.9741828 -0.00587789 0.04557631]

[-0.03027563 -0.00587789 0.97960271 -0.02036686]

[ 0.46288271 0.04557631 -0.02036686 0.9725373 ]]

1. With the same covariance Matrix and same samples as in Q4, we scan the most probable vector.

We expect to get the result of and this is indeed the point with the highest log-likelihood.

A screenshot of a computer

Description automatically generated with medium confidence

1. The highest probable are

(f1,f3) = (-0.05, 3.97)  
(Note it comes from an estimation with resolution of and at the center of the range we have ~ … -0.15, -0.05, 0.05, 0.15 … )