Exercises
Machine Learning
AS 2012

Machine Learning Laboratory
Dept. of Computer Science, ETH Zürich
Prof. Dr. Joachim M. Buhmann
Web http://ml2.inf.ethz.ch/courses/ml/

Email questions to: Morteza Haghir Chehreghani
morteza.chehreghani@inf.ethz.ch

Series 1, Sept 25st, 2012 (Statistics of Distributions)

Please turn in solutions until Tuesday, Oct 2th.

Problem 1 (Sampling a D-dimensional Gaussian distribution):

We consider the problem of sampling a multivariate normal (Gaussian) distribution. Matlab provides a function called `randn`, which produces pseudo-random samples for a normal distribution with parameters $\mu = (0, \ldots, 0)$ and $\mathbf{\Sigma} = \mathbf{1}$, where $\mathbf{1}$ is the D-dim. identity matrix. We wish to produce samples from a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ with arbitrary parameters $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, so we have to transform the sample in a suitable manner.

Our approach is based on the eigenvalue structure of symmetric matrices: The eigenvectors of a full-rank symmetric matrix form an orthonormal basis of the underlying vector space. With respect to this basis, the matrix is diagonal, with the eigenvalues as diagonal entries. Denote this diagonal matrix of eigenvalues $\mathbf{\Lambda}$ and the matrix describing the change of basis $\mathbf{U}$. Thus,

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} .$$

$\mathbf{U}$ is orthogonal (since it describes a change of basis between two orthonormal bases), so $\mathbf{U}^{-1} = \mathbf{U}^{\top}$ and $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}$. This representation of $\mathbf{\Sigma}$ is called the *Schur decomposition*.

We can produce a sample from a normal distribution with parameters $\boldsymbol{\mu}, \mathbf{\Sigma}$ by drawing a sample vector $\mathbf{g}$ from $\mathcal{N}(0, 1)$ using `randn`, changing basis, and adding the expectation vector:

$$\tilde{\mathbf{g}} = \mathbf{U}\sqrt{\mathbf{\Lambda}}\mathbf{g} + \boldsymbol{\mu} .$$

As you will recall from linear algebra, $\sqrt{\mathbf{\Lambda}} = \text{diag}\left(\sqrt{\mathbf{\Lambda}_{ii}}\right)$.

1. Implement a function `x = GSAMPLE(mu, Sigma, n)` to produce `n` draws from a D-dimensional Gaussian. (The dimension D is implicitly specified by `mu` and `Sigma`.)
2. For $\boldsymbol{\mu} = \begin{pmatrix} 5 \\ 10 \end{pmatrix}$, observe the following choices for $\mathbf{\Sigma}$:
   - $\mathbf{\Sigma}_1 = \begin{pmatrix} 4 & 2 \\ 1 & 4 \end{pmatrix}$
   - $\mathbf{\Sigma}_3 = \begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix}$
   - $\mathbf{\Sigma}_2 = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$

   For each choice of $\mathbf{\Sigma}$, explain if it is a valid choice for a covariance matrix.
3. Test your implementation on the above selection of matrices: Apply the matlab functions `mean` and `cov` for $n = 100, n = 1000$ and $n = 10000$ samples. What do you observe? How do the solutions approximate the input? In particular, for the wrong choices of $\mathbf{\Sigma}$, what has happened?
4. Produce 2000 samples each in two and three dimensions, using the parameter values $\boldsymbol{\mu} = (10, 10)^{\top}$, $\mathbf{\Sigma} = \begin{pmatrix} 10 & 4 \\ 4 & 5 \end{pmatrix}$ and $\boldsymbol{\mu} = (10, 10, 10)^{\top}$, $\mathbf{\Sigma} = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{pmatrix}$, respectively. Plot your results using the functions `plot` and `plot3`.

When using the `plot` function, always supply 'x' as final argument, i.e. use a function call of the form `plot(A,B,'x')`. (If your plot looks somewhat like a random walk, you got it wrong.) Please do not submit any code, instead report your numerical test results.

Problem 2 (Robust Estimation and Sample Size):

A recurrent theme in statistical data analysis is that large numbers of observations are required in order to obtain reliable estimates. The present problem aims at illustrating this phenomenon, using a Gaussian distribution as an example data source. We will draw a number n of sample points from a one-dimensional Gaussian, sort them into a histogram, and see how stable the result is with respect to different samples. The basic procedure is the following:

- Choose a number n of sample points and a number N_{bins} of histogram bins.
- Use the Matlab function `randn` to draw n samples from a one-dimensional Gaussian distribution ($\mu = 0, \sigma = 1$).
- Turn the data sample into a histogram using the function `hist`.

Please complete the following steps:

1. Produce four histograms with $n = 100$ and $N_{\text{bins}} = 10$. Plot the histograms using the `plot` function.
2. Repeat the procedure with $n = 100000$.
3. For $n = 100000$, plot one histogram each for $N_{\text{bins}} \in \{10, 100, 1000\}$.
4. Finally, choose $n = 100$ and $N_{\text{bins}} = 1000$.
5. Give a brief discussion of the results. Remember, this is about sample size and reliability of estimates.

Please turn in your plots and discussion. Make sure that plots are sufficiently labeled.