Kullback–Leibler divergence

In probability theory and information theory, the Kullback_Leibler divergence is a non-symmetric measure of the difference between two probability distributions P and Q. Although it is often intuited as a distance metric, the KL divergence is not a true metric. For example, the KL divergence from P to Q is not necessarily the same as the divergence from Q to P. For probability distributions P and Q of a discrete random variable, the KL divergence from Q to P is defined to be:

$$D_{KL}(P\|Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \quad (1)$$

On the other hand, if P and Q are probability distributions of a continuous random variable, the KL divergence from Q to P is defined as:

$$D_{KL}(P\|Q) = \int_{-\infty}^{\infty} P(x) log \frac{P(x)}{Q(x)} dx \quad (2)$$

Based on the equations (1) and (2), we can define a symmetric measure of the difference between the distributions P and Q as following:

$$D(P,Q) = \frac{D_{KL}(P\|Q) + D_{KL}(Q\|P)}{2} \quad (3)$$

Usually in practice we are given a data set X that corresponds to a repeated execution of the same experiment. Also we are given a set of parameterized probability distributions $P = \{P_1(x|\boldsymbol{\theta}_1), P_2(x|\boldsymbol{\theta}_2), \dots , P_N(x|\boldsymbol{\theta}_N)\}$. For example, $P_1(x|\boldsymbol{\theta}_1)$ could correspond to a family of Gaussian distributions. In this case, the parameter vector $\boldsymbol{\theta}_1$ contains the mean and the variance. For each family of distributions $P_i \in P$ we can compute the parameter vector $\boldsymbol{\theta}_i^*$ that fits better to the data set X using the method of maximum likelihood estimation [1]. Our goal is to determine which of the distributions $P_1(x|\boldsymbol{\theta}_1^*), P_2(x|\boldsymbol{\theta}_2^*), \dots , P_N(x|\boldsymbol{\theta}_N^*)$ describes better the dataset X.

To do this we first define the empirical probability distribution function Q that corresponds to the data set X. In particular we divide the probability space into a number of equally sized and non-overlapping bins. At each bin we assign a probability that is equal to the ratio of the number of samples that lie inside this bin versus to the total number of samples. We also perform a discretization of the probability distributions $P_1(x|\boldsymbol{\theta}_1^*), P_2(x|\boldsymbol{\theta}_2^*), \dots , P_N(x|\boldsymbol{\theta}_N^*)$. Specifically we assume that each $P_i(x|\boldsymbol{\theta}_i^*)$ is approximated by the discrete probability distribution function $\hat{P}_i(x|\boldsymbol{\theta}_i^*)$ that is constant within each bin and equal to the mean value of $P_i(x|\boldsymbol{\theta}_i^*)$ at the corresponding bin. Another way that we could use in order to estimate $\hat{P}_i(x|\boldsymbol{\theta}_i^*)$ more easily is to create a dataset Y of randomly drawn samples from the distribution $P_i(x|\boldsymbol{\theta}_i^*)$. The empirical distribution function that corresponds to the dataset Y is an estimation of $\hat{P}_i(x|\boldsymbol{\theta}_i^*)$.

Subsequently, for each pair $\left(Q,\ \hat{P}_i(x|\theta_i^*)\right)$ we compute the metric $D\left(Q,\ \hat{P}_i(x|\theta_i^*)\right)$ according to equation (3). Finally, the distribution with the minimum distance from Q is assumed to describe better the dataset X.

The above procedure is implemented by the function `kld_test_all_g_sym`. This function uses the following families of distributions:

Name	Parameters	PDF (probability distribution function)	CDF (cumulative distribution function)
Rayleigh	σ	$P(x) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}$	$C(x) = 1 - e^{-x^2/2\sigma^2}$
Lognormal	$\mu, \sigma > 0$	$P(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(lnx-\mu)^2}{2\sigma^2}}$	$C(x) = \frac{1}{2} erfc\left(-\frac{lnx-\mu}{\sigma\sqrt{2}}\right)$
Weibull	$\lambda > 0, \kappa > 0$	$P(x) = \frac{\kappa}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^\kappa}\ x \geq 0$	$C(x) = 1 - e^{-(x/\lambda)^\kappa}\ x \geq 0$
Gamma	$\kappa > 0, \theta > 0$	$P(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^\kappa \Gamma(\kappa)} \quad x \geq 0$	$C(x) = \frac{\gamma(\kappa, \chi/\theta)}{\Gamma(\kappa)} \quad x \geq 0$
Exponential	$\lambda > 0$	$P(x) = \lambda e^{-\lambda x} \quad x \geq 0$	$C(x) = 1 - e^{-\lambda x} \quad x \geq 0$
Generalized pareto	$k \neq 0, \sigma > 0, \theta$	$P(x) = \frac{1}{\sigma}\left(1 + \kappa\frac{x-\theta}{\sigma}\right)^{-1-1/\kappa}$	$C(x) = 1 - (1 + k\frac{x-\mu}{\sigma})^{-1/\kappa}$

The following table shows the matalb functions that implement the method of maximum likelihood estimation for various families of distributions:

Distribution	Matlab function
Rayleigh	rayfit
Lognormal	lognfit
Weibull	wblfit
Gamma	gamfit
Exponential	expfit
Generalized pareto	gpfit

The input parameters of the function `kld_test_all_g_sym` are the following:

Input parameters	Description
rd	A column vector that contains the samples of the dataset X.
Nbins	The number of bins that we use in order to compute the empirical probability distribution function.

Finally, the output parameters of `kld_test_all_g_sym` are:

Output parameters	Description
dist	A 6x2 matrix. The first column contains the mean value of the metric of equation (3) for all distributions in the sequence: Rayleigh, Weibull, Lognormal, Pareto, Gamma and finally Exponential. The second column contains the variance of the metric of equation (3).

Example

First we run the matlab command:

```
data = wblrnd(1, 1.5, 10000,1);
```

This command creates a vector of 10000 randomly drawn samples from the Weibull distribution with parameters $\lambda = 1$ and $k = 1.5$. Subsequently we execute the command:

```
dist = kld_test_all_g_sym(data, 50);
```

The components of the matrix dist are shown in the following table:

Distribution	Mean	Variance
Rayleigh	0.1158	0.0168
Weibull	**0.0092**	0.0021
Lognormal	0.1079	0.0057
Pareto	0.0670	0.0039
Gamma	0.0117	0.0019
Exponential	0.1457	0.0047

The minimum mean distance as expected corresponds to the Weibull distribution.

ccdf (complementary cumulative distribution function)

The complementary cumulative distribution function of a random variable X is given by the following relation:

$$F_c(x) = P(X \geq x)$$

The function $F_c(x)$ is also related with the cumulative distribution function of X by the following equation:

$$F_c(x) = 1 - P(X \leq x) = 1 - F(x)$$

Suppose that we want to check whether a given dataset X fits with a particular family of distributions $P(x|\boldsymbol{\theta})$. In this case we first use the method of maximum likelihood estimation to calculate the value of the parameter vector $\boldsymbol{\theta}^*$ that describes better the dataset X. Then we plot the empirical ccdf of the dataset X and the ccdf of the distribution $P(x|\boldsymbol{\theta}^*)$. If these two ccdf curves are close we can conclude that the dataset X is likely to be generated by the distribution $P(x|\boldsymbol{\theta}^*)$.

The name of the matlab function that implements the ccdf test is plotccdf. The input parameters of this function are the following:

Input parameters	Description
data	A column vector that contains the samples of the dataset X.
distribution	A string which can take one of the following values: ‘weibull’, ‘lognormal’, ‘exponential’, ‘pareto’, ‘gamma’, ‘rayleigh’. Each value corresponds to a different family of distributions.
log	If the value of this parameter is 1 the ccdfs are plotted in log-log scale. Otherwise they are plotted in linear scale.

Example

We first create a dataset of 10000 samples drawn from a weibull distribution with parameters $\lambda = 1$ and $\kappa = 1.5$ using the following command:

```
data = wblrnd(1, 1.5, 10000,1);
```

Then we execute the command:

```
ccdf_test( data , 'weibull', 0);
```

The result is shown in figure 1. From this figure we can easily observe that that dataset X is described well by a Weibull distribution as expected.

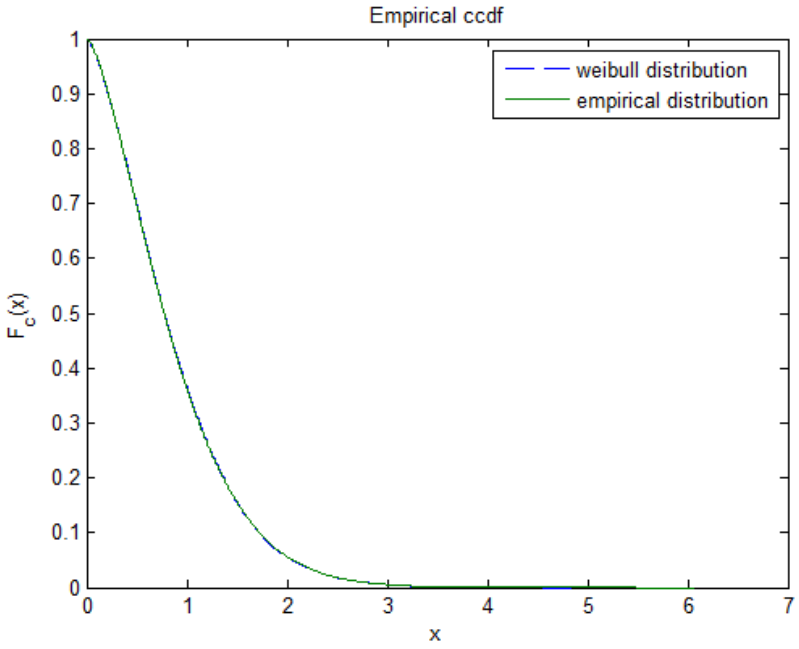

Figure 1: Plot that is produced by the execution of the command ccdf_test(data , 'weibull', 0);

Q-Q plots

In statistics, a **Q-Q plot** is a probability plot, which is a graphical method for comparing two probability distributions by plotting their percentiles against each other. A percentile is the value

of a variable below which a certain percent of observations fall. For example, the 20th percentile is the value, below which 20 percent of the observations may be found.

In order to produce the Q-Q plot that corresponds to two different datasets X and Y we first discretize the interval [0 1]. In particular, we take a number of samples $s(1), s(2), \dots, s(N)$ that are equally spaced and cover the entire range of values from 0 to 1. In figure 2 we show the discretization of the interval [0 1] into 11 samples.

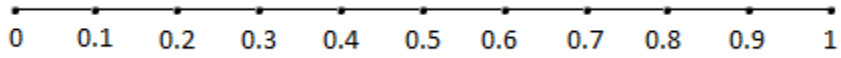

Figure 2: Discritization of the interval [0 1] to 11 equally spaced samples.

Subsequently we evaluate the inverse of the cumulative distribution function that corresponds to the dataset X at all the samples s(1), …, s(N). The values that are computed this way are the percentiles of the datasets X and are denoted as $p_X(1), p_X(2), \dots, p_X(N)$. Similarly we can compute the percentiles $p_Y(1), p_Y(2), \dots, p_Y(N)$ of the dataset Y. If we plot the percentiles of Y versus the percentiles of X, we get the Q-Q plot. In figure 3 we show how we calculate the percentiles of a dataset X of 10000 samples drawn from an exponential distribution of mean $\lambda = 2$ and the percentiles of a dataset Y of 10000 samples drawn from the gamma distribution with parameters $k = 2.5$ and $\theta = 2.5$.

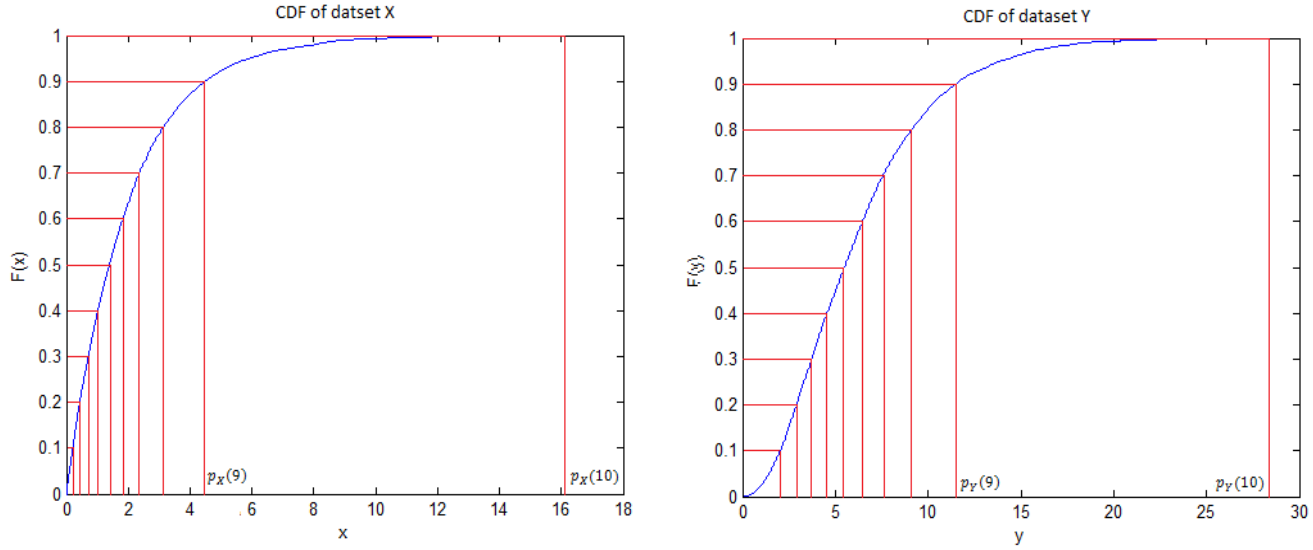

Figure 3: The left diagram shows the procedure that we follow in order to calculate the percentiles of a dataset X of 10000 samples drawn from an exponential distribution of mean $\lambda = 2$. The right diagram shows he same procedure for a dataset Y of 10000 samples drawn from a gamma distribution with parameters $k = 2.5$ and $\theta = 2.5$.

In order to check if a dataset X is generated from a particular family of distributions $p(x|\boldsymbol{\theta})$ we first use the maximum likelihood estimation method to compute the parameter vector $\boldsymbol{\theta}^*$ that fits better to the dataset X (maximizes the log-likelihood function). Subsequently we generate a number of datasets $Y_0,\ Y_1,\ Y_2, \dots, Y_N$ of the same size as X with samples that are drawn from the probability distribution $p(x|\boldsymbol{\theta}^*)$. Then we plot the percentiles of each of the datasets $Y_1, Y_2, \dots, Y_n$

versus the percentiles of Y_0. That way we construct the envelope of the distribution $p(x|\boldsymbol{\theta}^*)$. Finally we plot the percentiles of X versus the percentiles of Y_0. If this final curve lies within the envelope of $p(x|\boldsymbol{\theta}^*)$ we can conclude that the probability distribution $p(x|\boldsymbol{\theta}^*)$ describes well the dataset X.

The Q-Q plot test that was described above is implemented by the function envelope_qqplot. The input parameters of this function are the following:

Input parameters	Description
indata	A column vector that contains the samples of the dataset X.
distribution	A string which can take one of the following values: 'weibull', 'lognormal', 'extreme_value', 'exponential', 'generalized_extreme_value', 'pareto', 'gamma', 'rayleigh' and 'bipareto'. Each value corresponds to a different family of distributions.
samples	The number of synthetic datasets that will be used to create the envelope.

The output parameters of the function envelope_qqplot are the following:

Output parameters	Description
paramhat_vec	The value of the parameter vector $\boldsymbol{\theta}^*$ that is calculated using the method of maximum likelihood estimation.

Example

We first create a dataset X of 10000 samples drawn from a Weibull distribution with parameters $\lambda = 1$ and $\kappa = 1.5$ using the following command:

```
data = wblrnd(1, 1.5, 10000,1);
```

Then we run the command:

```
[paramhat_vec] = envelope_qqplot(data, 'weibull', 50);
```

The value of the parameter paramhat_vec that is returned by the function envelope_qqplot is equal to [0.9912 1.4905] which is very close to the actual parameters of the Weibull distribution. Also the Q-Q plot that is created by the function envelope_qqplot is shown in figure 4. The blue points correspond to the original data while the green points correspond to the envelope. We can easily observe that the blue points lie within the envelope which means that the dataset X is described well by a weibull distribution as expected.

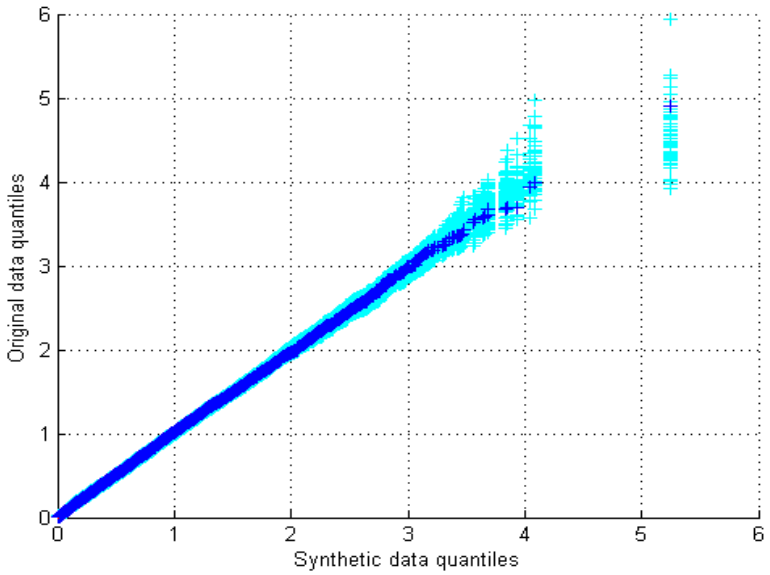

Figure 4: The Q-Q plot that is produced by the execution of the command [paramhat_vec] = envelope_qqplot(data, 'weibull', 50);

References:

[1] “Pattern Classification,” R. O. Duda, P. E. Hart, D. G. Stork, Wiley- Intersience, Second Edition, New York 2001.