Fitting data into probability distributions

Tasos Alexandridis

analexan@csd.uoc.gr

Problem statement

- Consider a vector of N values that are the results of an experiment.
- We want to find if there is a probability distribution that can describe the outcome of the experiment.
- In other words we want to find the model that our experiment follows.

Probability density function: $f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

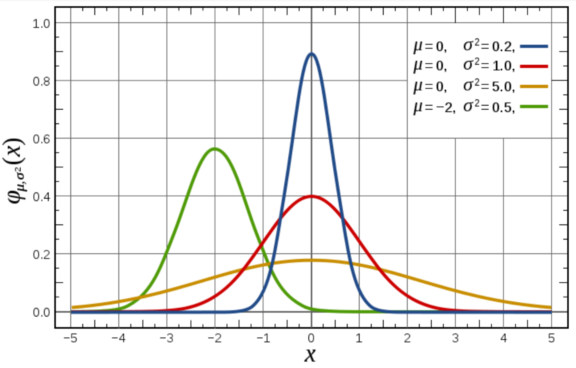


Figure: The Gaussian distribution

The red line is the *standard normal distribution*

Probability density function: $f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, x \geq 0 \\ 0, x < 0 \end{cases}$

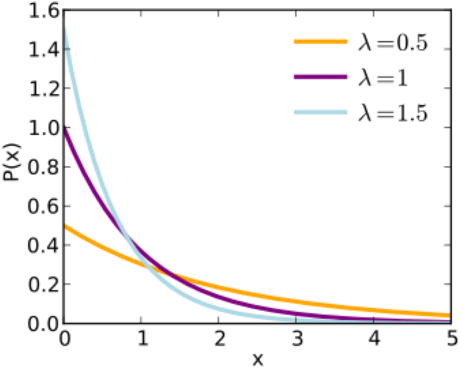


Figure: The exponential distribution

Probability distributions: *The exponential distribution (cont)*

Exponentially distributed random variables are memoryless

$$P\{X > s + t | X > t\} = P\{X > s\}$$

If we think X as being the lifetime of some instrument, then the probability of that instrument lives for at least s+t hours given that it has survived t hours is the same as the initial probability that it lives for at least s hours.

In other words, the instrument does not remember that it has already been in use for a time t

Probability density function: $f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}$

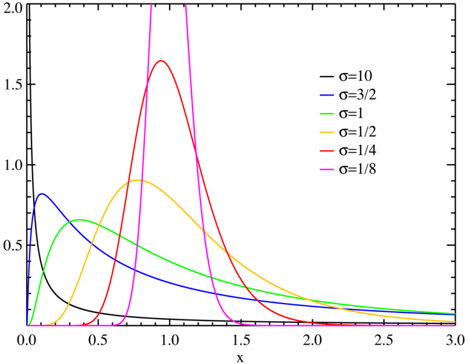


Figure: The lognormal distribution

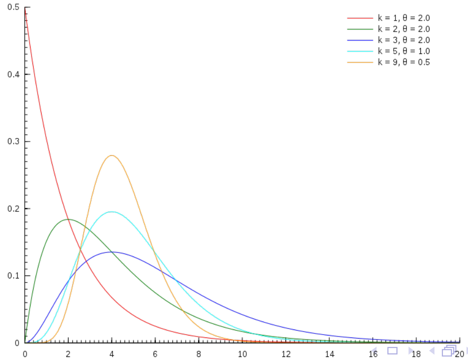
The lognormal distribution is a probability density function of a random variable whose logarithm is normally distributed

Probability density function:

$$f(x;\alpha,\beta)=\begin{cases}\frac{\lambda e^{-\lambda x}(\lambda x)^{\alpha-1}}{\Gamma(a)}, ifx \geq 0\\ 0, ifx < 0\end{cases}$$

The quantity $\Gamma(a)$ is called Gamma function and is given by:

$$\Gamma(\alpha)=\int_0^{\infty} e^{-x}x^{\alpha-1}dx$$



Probability density function: $f(x;\sigma) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, x \geq 0$

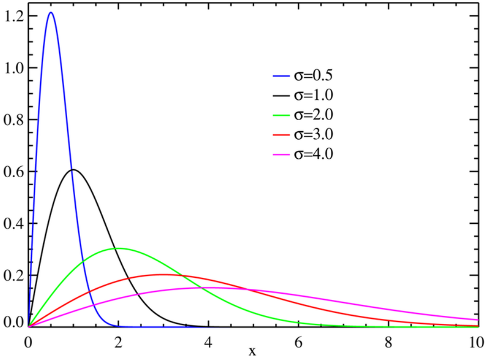


Figure: The rayleigh distribution

Example: Random complex variables whose real and imaginary parts are i.i.d. Gaussian. The absolute value of the complex number is Rayleigh-distributed

Counting processes

A stohastic process $\{N(t), t \geq 0\}$ is said to be a *counting process* if N(t) represents the total number of "events" that have occured up to time t. A counting process must satisfy:

- $N(t) \geq 0$
- N(t) is integer valued.
- If $s < t$ then $N(s) \leq N(t)$
- For $s < t$, N(t)-N(s) equals the number of events that have occured in the interval (s,t)

Poisson process

A counting process $\{N(t), t \geq 0\}$ is said to be a Poisson Process having rate $\lambda, \lambda > 0$, if

- N(0) $= 0$
- The process has independent increments i.e. the number of events which occur in disjoint time intervals are independent.
- The number of events in any interval of length t is Poisson distributed with mean λt. That is, for all $s, t \geq 0$:

$$P\{N(t+s) - N(s) = n\} = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, n = 0, 1, ...$$

Poisson process. Interarrival time

Consider a Poisson Process, and let us denote the time of the first event by T1. Further, for$n > 1$, let Tn denote the time elapsed between the (n-1)st and the nth event. The sequence { Tn, n = 1,2,... } is called *sequence of interarrival times*.

Example: If T1 = 5 and T2 = 10, then the first event of the Poisson process whould have occured at time 5 and the second event at time 15

Proposition $T_n, n = 1, 2...$, are independent identically distributed exponential variables. (i.e. the interarrival times of a Poisson Process are exponentially distributed)

Fitting procedure: Overview

- Fit your real data into a distribution (i.e. determine the parameters of a probability distribution that best fit your data)
- Determine the goodness of fit (i.e. how well does your data fit a specific distribution)
    - qqplots
    - simulation envelope
    - Kullback-Leibler divergence

Example: Fitting in MATLAB

Generate data that follow an exponential distribution with $\mu = 4$

```
values = exprnd(4,100,1);
```

Generate random Gaussian noise N(0,1)

```
noise = randn(100,1);
```

Add noise to the exponential distributed data so as to look more realistic

```
real_data = values + abs(noise);
```

Consider `real_data` to be the values that you want to fit

```
[paramhat] = expfit(real_data);
 >> 4.9918
```

The estimated μ parameter is 4.9918

In other words, our data fit an exponential distribution with $\mu = 4.9918$

Example: Fitting in MATLAB
Test goodness of fit using qqplot

Generate synthetic data from the probability distribution you found to fit your real data and plot the real versus the sythetic data

The closer the points are to the y=x line, the better the fit is.

```
syntheticData = exprnd(4.9918,100,1);
qqplot(real_data,syntheticData);
```

Example: Fitting in MATLAB
Test goodness of fit using qqplot

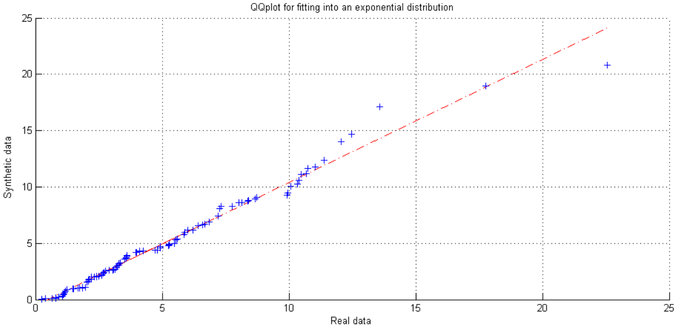


Figure: QQplot for fitting into an exponential distribution

Example: Fitting in MATLAB
Test goodness of fit using simulation envelopes

- Fit your data into the specified distribution.
- Create synthetic data (wdata0)
- Run a number of N tests . For every test i
  - Create synthetic data
  - Make the qqplot of wdata0 and the synthetic data created for test i
- An "envelope" will be created
- Finally make the qqplot of the the real data and wdata

For a "good" fit the qqplot of the real data, should be inside the envelope

Example: Fitting in MATLAB
Test goodness of fit using simulation envelopes

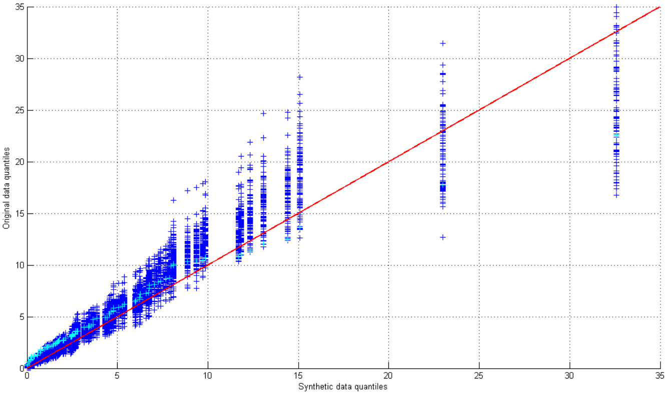


Figure: Simulation envelope for exponential fit with 100 runs

Kullback-Leibler Divergence or **Relative Entropy** between two probability mass vectors p and q

$$D(p||q) = \sum_{x \in X} p(x) log \frac{p(x)}{q(x)}$$

- $D(p||q)$ measures the "distance" between the probability mass function p and q
- We must have $p_i = 0$ whenever $q_i = 0$ else $D(p||q) = \infty$
- $D(p||q)$ is not the true distance because:
  1. it is assymetric between p and q i.e. $D(p||q) \neq D(q||p)$
  2. it does not satisfy the triangle inequality