Statistical Distribution Identification with Cloud Based Module

Dr. Ventsislav Nikolov
Senior Software Developer
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: vnikolov at eurorisksystems dot com

Danko Naydenov
Senior Software Developer
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: sky at eurorisksystems dot com

Dr. Anatoliy Antonov
CEO
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: antonov at eurorisksystems dot com

Abstract: *In this paper an implemented software system for identification of best fitting distribution of sample data is described. Some modifications and additions of the known statistical approaches are presented aiming the practical application of the distribution identification task. Additionally, the cloud computing approach is applied in order to process the sample data series in parallel that makes significantly faster the implemented system.*

Keywords*: Best Fit Distribution; Distribution Parameters; Cloud Computing*

I. INTRODUCTION

Distribution type identification from empirical data (fitting a probability distribution to data) is one of the main tasks in the statistics. Generally, there are different approaches solving this task with different results according to the nature of the sample data. In this paper two alternative modifications of known approaches are described implemented by the authors of this paper in a software library for distribution type identification. The library works based on cloud computing making concurrent calculations for many data samples. Below first the cloud computing features are described and after that the distribution identification approaches are presented. Finally the performance and characteristics of the implemented software system are described and shortly analyzed.

II. TRANSITION OF A TRADITIONAL APPLICATION TO CLOUD COMPUTING

The organization of the distribution type identification system is shown on Fig.1. The process of distribution determination can be implemented in parallel because every sample is considered as independent from the others. However, if there are N data samples processed in parallel this does not accelerate N times the processing time compared to the sequential processing. Ignoring the delays caused by the communication or synchronization of the parallel processes the acceleration generally depends on the number of parallel processors solving the task.

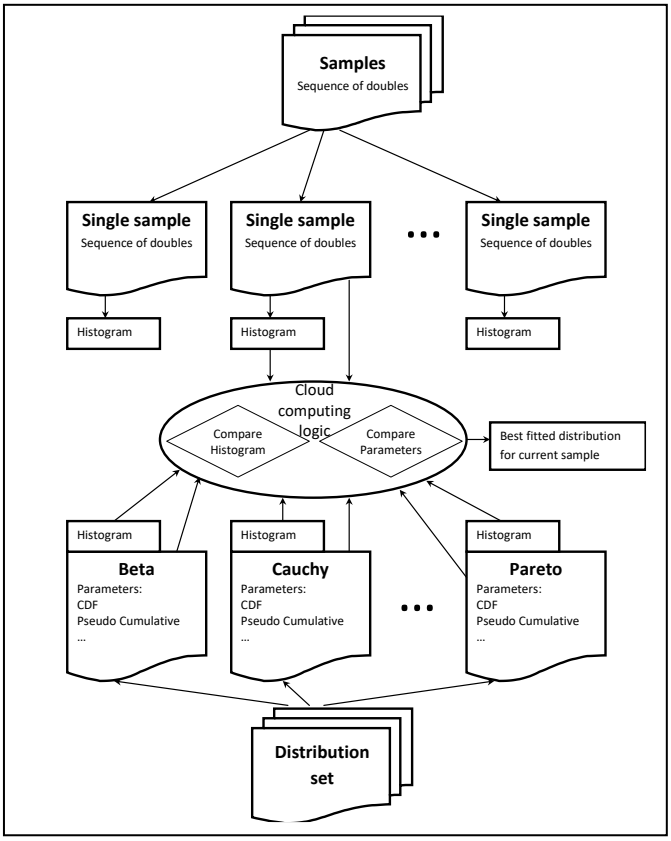

Fig. 1. The cloud based system for distribution identification

The most computer systems today are supplied with more than one processor or physical threads but if there are more running tasks than physically available processors then the time

slicing approach should be applied by the operating system. This generally could accelerate the processing not more than the number of processors is in the system.

Unlike a particular computer configuration, where the hardware parameters are static the cloud environment is dynamic and can expand the capacity of the existing resources. This makes it a better alternative in the case of parallel distribution type identification. The cloud approach will allocate a separate processor for each series and this makes the acceleration to be equal to the number of processed series. The communication time should also be taken into account because in the local system software processes communicate fast enough but in the cloud computing the communication may cause a significant delay. So the cloud based computing also requires addressing the communication issues.

As a cloud provider the Google App Engine is chosen. The reason for that is because this is a Platform as a Service (PaaS) provider which offer automatic scaling and support application written in Java. This type of service is able to handle only HTTP requests. That is why the migration of the existing software library to the cloud application requires either specific software reorganization as a web service or explicit data serialization/deserialization in the data transfer stage.

The first approach is not convenient because the cloud platforms are still not developed enough to provide a good support for web services. The automation degree of the services in cloud environment is low which leads to the need of efforts to create a Web Services server. The second approach is easier to implement but in this case a direct serialization cannot be applied because the HTTP is a text based protocol. The data objects must be serialized to a text format. XML serialization has the disadvantage that in addition to the transferred data there is also a large amount of formatted data tags. In order to avoid these problems the more suitable JSON serialization format is used here. It is becoming increasingly popular in the web applications because of the direct relation to JavaScript and the fact that the most modern browsers have built-in support for it. Moreover, in opposite to the XML, this format adds very little overhead in the message. For example, an object of a class Person, that contains two fields respectively for the first and last name, is transformed to a XML message like this:

```
<person>
   <first-name>George</first-name>
   <last-name>Brown</last-name>
</person>
```

For the same object using JSON format the following text is generated:

```
{Person:{"first-name":"John","last-name":"Smith"}}
```

III. DISTRIBUTION TYPE IDENTIFICATION

The best fitting distribution type of a sample data provides a useful theoretical description of the inherent data properties. For example, the identified best distribution type can be used for the VaR estimation in Copula Monte Carlo simulation [4, 7]. Calculating the percentile as part of the VaR estimation assumes that the sample data is normally distributed – Fig.2. However, for the practically obtained data samples this is not always true.

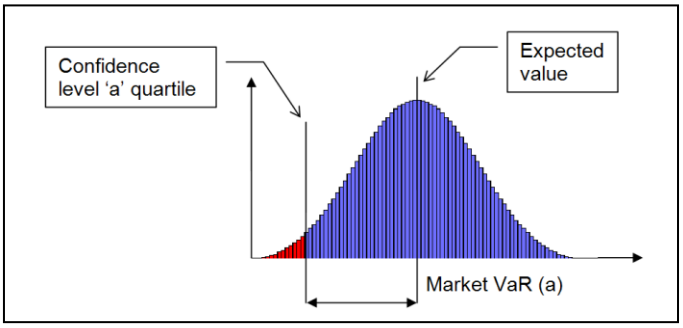

Fig. 2. VaR estimation

The identification of the true (or better than Normal for the given sample data) distribution type in this case could produce more accurate VaR estimation because of the different distribution shape tails [2].

The hypothesis goodness of fit test generally indicates whether or not the sample belongs to a preliminary specified distribution type [6, 9]. Examples of such methods are "chi squared", Kolmogorov-Smirnov, Anderson-Darling, etc. In our case the best distribution type is determined choosing from a list of specified beforehand distributions and this could be done by different approaches. Here two alternative measures are used.

A. Histogram measure.

In this case the average squared distance between the histogram bin frequencies is calculated (2). This type of measure is similar to the "chi squared" goodness of fit test statistic (1). It requires building of empirical and theoretical histograms and working on binned data.

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \quad (1)$$

$$d^2 = \frac{1}{k} \sum_{i=1}^{k} (O_i - E_i)^2 \quad (2)$$

The empirical histogram can be easily built from the sample data but the theoretical histogram uses the cumulative distribution function (CDF) for every distribution type. This means that the distribution parameters for every distribution type should be estimated beforehand. In table I the implemented in the presented system distributions types and their parameters are shown.

There are one or two distribution specific parameters and one or two additional parameters.

- Distribution specific parameters. For each distribution type its specific parameters are estimated from the empirical sample data using the method of moments, least squares regression or the maximum likelihood estimation [1, 3]. In the most cases there are more than one possible ways for estimation. The choice in this case is based on the known researches about their effectiveness [8, 3].
- Additional parameters. From practical point of view it is possible the sample data, or part of it, to fall in region in which a distribution is not defined. Taking into account that the main object is to identify the distribution based on the probability distribution curve shape the data is shifted and scaled in order to compare the empirical to the theoretical shape – fig.3.

TABLE I. DISTRIBUTIONS AND THEIR PARAMETERS

Distribution	Distribution Parameters		Additional Parameters	
	Parameter 1	Parameter 2	Parameter 1	Parameter 2
Beta	Shape	Shape	Shift	Scale
Cauchy	Location	Scale	---	---
Exponential	Rate	---	Shift	---
Inverse Normal	Mu	Lambda	Shift	---
Log Normal	Log Scale	Shape	Shift	---
Normal	Mean	Variance	Shift	---
Pareto	Scale	Shape	Shift	---
Rayleigh	Sigma	---	Shift	---
Student	Nu	---	Shift	---
Weibull	Scale	Shape	Shift	---

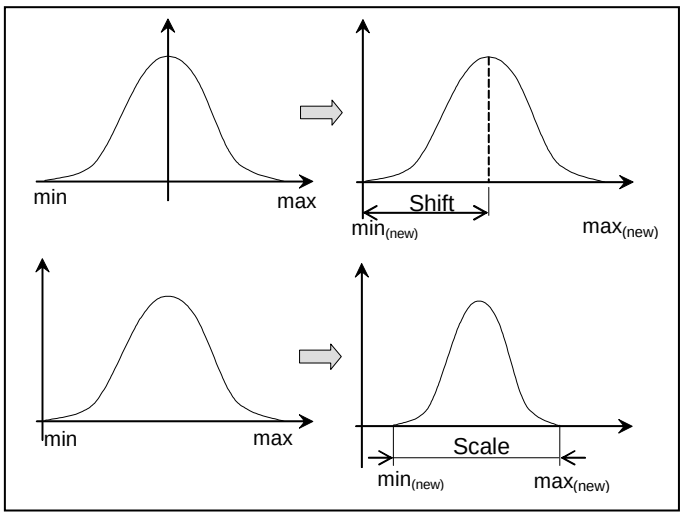

Fig. 3. Additional parameters

After obtaining both the distribution specific and the additional parameters the theoretical histogram is built. The starting point is the known CDF:

$$F_x(x) = P(X <= x) \quad (3)$$

The theoretical histogram is built as the probability distribution function (PDF) from a distribution with known CDF – fig.4.

$$P(a < X <= b) = F_x(b) - F_x(a) \quad (4)$$

where Fx(x) is the CDF.

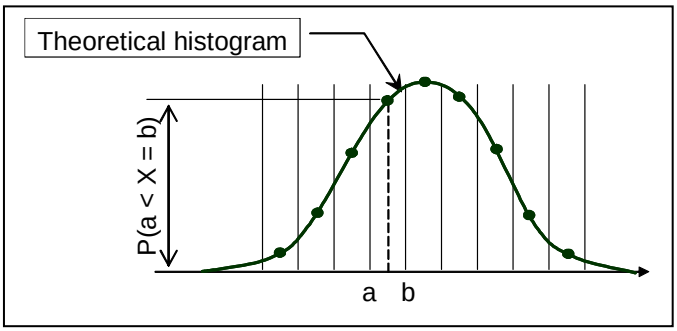

Fig. 4. Theoretical histogram bins

The empirical histogram is built by counting the values belonging to the bins. If a value is on a boarder between two bins then in the both of them a half a value is added. The number of bins of the theoretical histogram coincides with that for the empirical histogram. There are different ways to determine the number of bins. In our implementation the following formula is used:

$$B = 5\log(N) \quad (5)$$

and

$$B < 5 \Rightarrow B = 5 \quad (6)$$

$$B > 25 \Rightarrow B = 25 \quad (7)$$

where N is the observations count (sample size).

B. Cumulative measure.

In this case the cumulative distribution values are used. This is a completely new approach aiming to increase the accuracy of the best distribution identification taking into account all values in the distribution identification process. The cumulative values approach works on non-binned data and in this case more operations are needed leading to greater time consummation. The method is as follows:

- Having the sample values in the Y axis (shown in the left in fig.5) the differences between them are

accumulated and the graphic of this function is considered.

- A shifting is performed on the function in order its mean to coincide to the mean of the identical empirical data function.

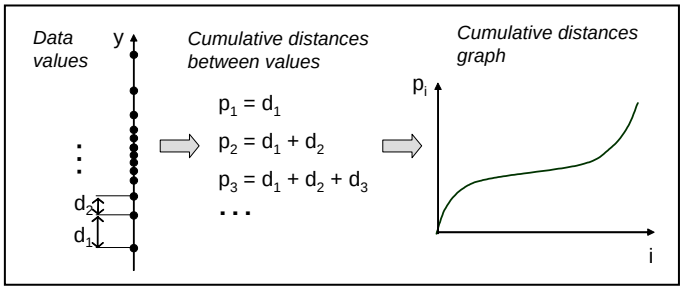

Fig. 5. Cumulative distances

The most challenging task using this approach is obtaining the sample with the theoretical distribution corresponding to the characteristics of the empirical sample data. Here again, as in the histogram measure, the theoretical parameters estimations are used. The full theoretical sample is generated using the inverse CDF (8).

$$x = F^{-1}(\xi) \tag{8}$$

The same sample values generator is also used to test the distributions identification system. The generation of the sample, distributed according to a specified distribution type, is performed by using of uniformly distributed values in the interval [0-1] generated either randomly or in equal distances [5].

The upper two approaches are programmatically implemented as visitor-like pattern that performs the corresponding calculation over the data structures. This pattern allows additional distribution identification methods to be easily added.

IV. RESULTS

To test the implemented approaches a prototype desktop java application is developed, shown on fig. 6, which allows the distributions to be ordered according to the goodness of fit. Thus, for example, only best 3 or 5 of them can be accepted and if the sample is classified to best fit a given distribution type but another distribution is also good enough then it could be manually changed. Both the histogram and cumulative measures are almost equal in the correctness of the distribution type identification. A disadvantage of the histogram measure is that it needs sufficient data. Moreover, it is sensitive to the chosen number of bins. The cumulative values method is better when sample data is not sufficient.

The execution time of the system using the cloud approach should also be considered. In fig.7 the time is shown for the identification of different number of threads locally and in the cloud. A thread is used for a distribution type identification of a data sample. As it can be seen in fig.7 with the increasing of the data samples number, where each sample is processed by a dedicated thread, the cloud based system is becoming much faster than the local system.

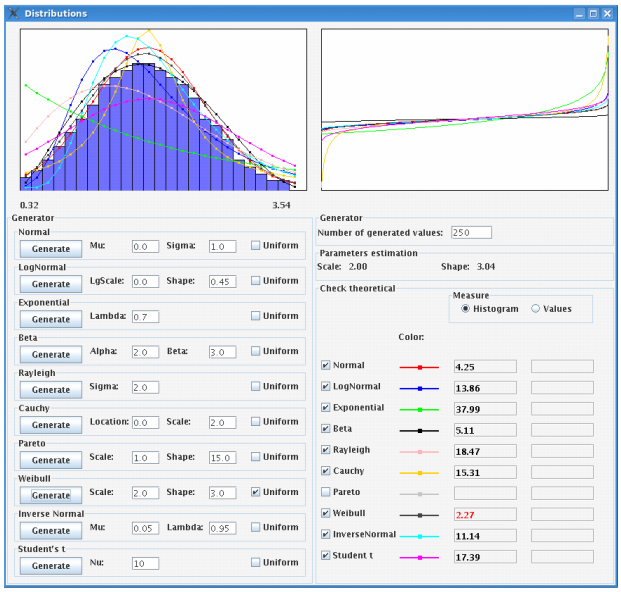

Fig. 6. The prototype of the distribution identification system

Although PaaS provides only environment for executing an application, the Google App Engine provides some basic parameters which can be configured and for the purpose of the experiment the most powerful configuration for CPU is selected – 2.4GHz, the memory is selected to be 512MB and these parameters are provided for each request to developed application. In other words, if there are twenty simultaneous requests then the cloud environment will provide for each of them separate CPU and RAM. The local system, which results are compared to the cloud based system, is with a two core CPU with frequency 2.4GHz and 2GB memory.

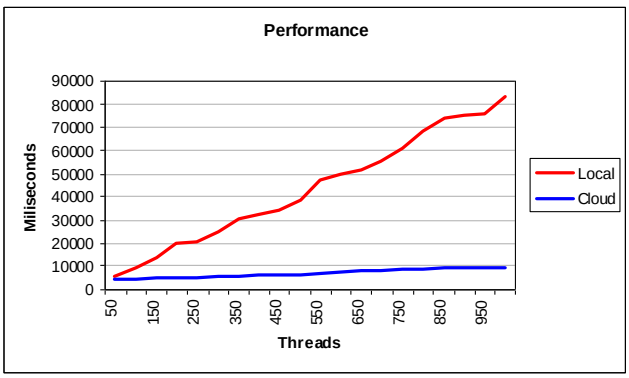

Fig. 7. Performance of the local and cloud based missing values system

V. CONCLUSIONS AND FUTURE WORK

The practical applications of theoretical approaches generally require much pre-processing and post-processing techniques in order to be usable in the real world. In the presented methods the cumulative values function shifting, the additional distribution parameters (shift and scale), and the other techniques aim to produce practically useful recognition of the probability distribution shape based on the pre-specified distribution types.

Using of the cloud based approach makes such kind of applications modern in the sense of the latest computer technologies. Thus, they are much faster and more convenient than the traditional desktop systems.

The presented software system could be extended to use additional distribution types as well as alternative distribution type identification approaches.

REFERENCES

[1] Balakrishnan, N., Nevzorov, V. A primer on statistical distributions. John Wiley & Ssons Inc., 2003.

[2] Bury, K. Statistical Distributions in engineering. Cambridge University Press, 1999.

[3] Krishnamoorthy, K. Handbook of statistical distributions with applications. Chapman & Hall, 2006.

[4] Mun, J. Modeling Risk. Applying Monte Carlo Simulation, Real Options Analysis, Forecasting, and Optimization Techniques. Wiley, 2006.

[5] R-forge distributions Core Team. A guide on probability distributions. 2009.

[6] Ricci, V. Fitting Distributions with R. 2005.

[7] Robert, C., Casella, G. Monte Carlo statistical methods. Springer-Verlag, 1999.

[8] Vallentin, M. Probability and Statistics Cookbook, 2011.

[9] Vose, D. Fitting distributions to data and why you are probably doing it wrong. 2010. http://www.vosesoftware.com.