ANALYSIS OF OIL RESERVES OF THE FIELD USING BUFFERED PROBABILITY OF EXCEEDANCE

A.A. Tokaeva[1]

[1] Moscow State University, Moscow, Russia, galynka@ymail.com

Abstract

In this paper we suggest a new method to evaluate oil reserves of the oil field by means of a new concept called "Buffered probability of exceedance" (bPOE) and its corresponding threshold called "Conditional Value at Risk" (CVaR). Through averaging of data in the tail, bPOE compactly presents probability as well as loss value of the tail and has the capability to revolutionize the concept of risk-averse engineering. It was first applied[1] to the problem of tropical storm damage analysis by Uryasev (2018) and then its properties were strictly proved[2] by Mafusalov and Uryasev (2016). Herein we not only suggest a bPOE-based approach to the problem of oil reserves calculation, but also compare the obtained results with those provided by the POE-based approach (and its threshold Value at Risk (VaR)) and make some useful remarks about reserves distribution.

1. Aims

Let's consider an oil field at the state of exploration, when some investigation has already been undertaken, and we have measurements for all coefficients that determine the initial oil reserves. Oil reserves are calculated as follows:

$$Q_0 = S * h * k * S_0 * B_0 * \rho$$

where S is the area of the field, h is effective oil-saturated thickness, k is porosity, S_0 is coefficient of oil saturation, $B_0 = 1.623$ is volume coefficient of reservoir oil, and $\rho = 0.8164$ is oil density. For each of these parameters we have already obtained numerical values in different wells of the field, then utilizing these measurements we reconstructed distributions of these parameters and now we have two goals:

1) Obtain some reliable estimation of oil reserves in this field.
2) Determine which parameter has more influence on the resulting reserves in order to apply such investigation methods that will allow us to clarify the values of this important parameter.

2. Methods

In the simplest model we assume that undefined parameters S, h, S_0 are normally distributed and k is lognormally distributed (with mean and standard deviation reconstructed using observations of these parameters) .

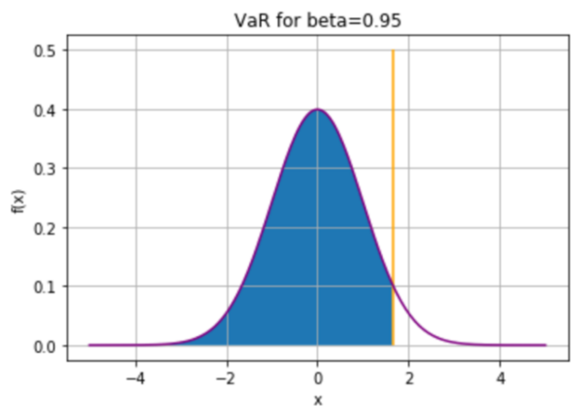
1) The baseline estimation for each parameter (and hence for oil reserves) is just the mean value of the distribution. But this solution of the problem cannot be adopted in business cases, because in some cases real values will be less than the mean, and in some they will be greater, and we doesn't have a faintest idea of how big or small they may be.

2) The second-best idea is to use VaR and POE. Let's denote our desired parameter by a random variable X (with corresponding normal distribution). We are interested in knowing the amount of our parameter (X) relative to some given threshold (W). The probability of exceedance (POE) for that threshold W is defined by

$$p_W(X) = P[X > W]$$

The threshold W in $p_W(X)$ is commonly referred to in economics and finance as the "Value at Risk" (VaR). For example, if $X \sim N(0,1)$, and $W = 0.95$, then $p_W(X) \approx 1.65$.

```
z_beta = 1.6448536269514722
```

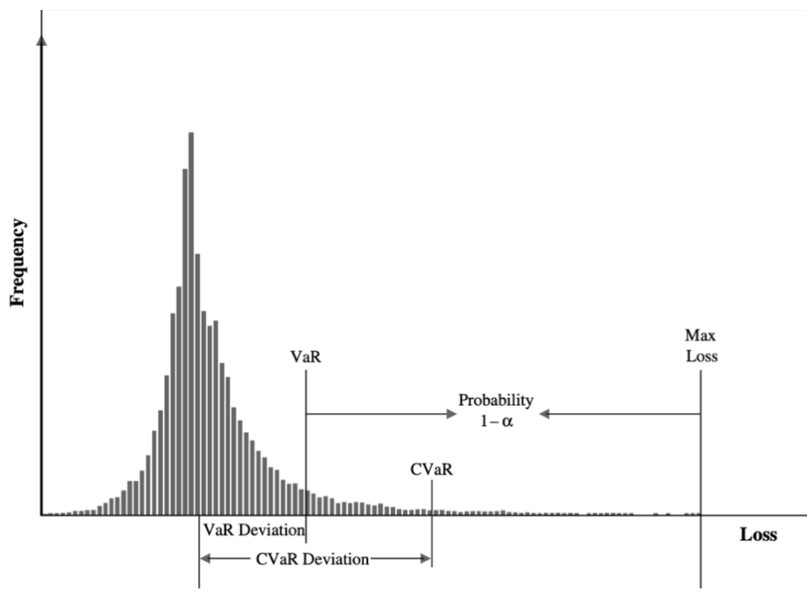


In other words, there is $\beta = 5\%$ probability that our parameter will exceed the VaR. But the flaw of this approach is that W and $p_W(X)$ do not provide any information about the magnitude of damages of X exceeding W.

3) The situation can be improved by using CVaR (and bPOE) instead of VaR (and POE). In order to introduce CVaR, we pose the following question: if the value of our parameter exceeds W, what is the average amount of this value expected? We say that the economic value of our parameter

$$V = E[X|X > W]$$

shows on average values exceeding W, where $E[X|X > W]$ is the conditional expectation that X exceeds W. By definition, V has a bPOE equal to $\bar{p}_V(X) = p_W(X)$. The threshold V in $\bar{p}_V(X)$ is commonly referred to in economics and finance as the "conditional value at risk" (CVaR). Note that CVaR is greater than VaR, so CVaR gives a more optimistic estimation. Similarly, in CVaR can be used the lower tail of the distribution instead of the upper tail. And then we will acquire information about the distribution of our parameter, if it is less than VaR. We will denote CVaR for the upper tail as $\overline{CVaR}$, and CVaR for the upper tail as $\underline{CVaR}$.



3. Results

Let's assume that our parameters have the following mean and standard deviation.

	P10	P50	P90	type
k	2.8	9.5	16.2	lognorm
h	9.3	12.6	15.9	norm
so	9.3	12.6	15.9	norm
s	440.0	460.0	480.0	norm

```
def find_mu_sigma(p10,p90,mytype):
    if (mytype=='norm'):
        return (p90+p10)/2,(p90-p10)/(scipy.stats.norm.ppf(0.9,loc=0,scale=1)-scipy.stats.norm.ppf(0.1,loc=0,scale=1
    elif (mytype=='lognorm'):
        return (np.log(p90)+np.log(p10))/2,(np.log(p90)-np.log(p10))/(scipy.stats.norm.ppf(0.9,loc=0,scale=1)-scipy.
    else:
        print("bad mytype")
```

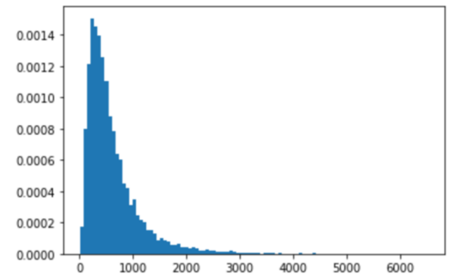
```
dfMuSigma=pd.DataFrame(columns=['mu','sigma','mytype'])
for i in range(len(df1)):
    tek_mu,tek_sigma=find_mu_sigma(df1.iloc[i,0],df1.iloc[i,2],df1.iloc[i,3])
    #print(i,tek_sigma)
    dfMuSigma=dfMuSigma.append({'mu':tek_mu,'sigma':tek_sigma,'mytype':df1.iloc[i,3]}, ignore_index=True)
dfMuSigma
```

	mu	sigma	mytype
0	1.907315	0.684870	lognorm
1	12.600000	2.575004	norm
2	12.600000	2.575004	norm
3	460.000000	15.606083	norm

Then we we generate $N = 10000$ realizations for each parameter and obtain the following distribution for the reserves.

```
N=10000
mas=[]
for i in range(N):
    tek=1
    for j in range(len(dfMuSigma)):
        if (dfMuSigma['mytype'][j]=='lognorm'):
            x=stats.norm.rvs(loc=dfMuSigma.iloc[j,0], scale=dfMuSigma.iloc[j,1], size=1)[0]
            x=np.exp(x)
        elif (dfMuSigma['mytype'][j]=='norm'):
            x=stats.norm.rvs(loc=dfMuSigma.iloc[j,0], scale=dfMuSigma.iloc[j,1], size=1)[0]
        #x=stats.lognorm.rvs(loc=0, scale=np.exp(dfMuSigma.iloc[j,0]), s=dfMuSigma.iloc[j,1], size=1)[0]
        tek*=x
    mas.append(tek)

mas=np.array(mas)
#mas=mas[mas>0]
plt.hist(mas/1000,bins=100,density=True)
plt.show()
```

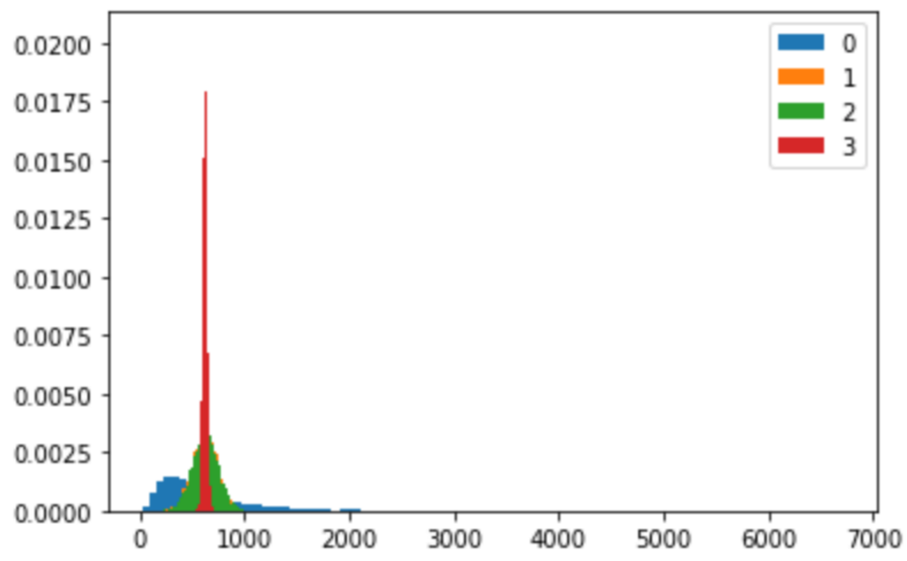


Now for this distribution we can calculate numerically $\overline{CVaR}$ and $\underline{CVaR}$ with some fixed POE in order to obtain information about average reserves in best and worst POE% of scenarios. This strategy gives us the solution to the first abovementioned question.

The second question was about the detection of most influential parameter in respect to amount of reserves. To answer it, we need to vary one parameter with fixed other parameters, calculate the difference between $\overline{CVaR}$ and $\underline{CVaR}$ for the obtained distribution of reserves and pick the largest.

```
dfOTV=pd.DataFrame(columns=['upper','lower'])
N=10000
for i in range(len(dfMuSigma)):
    init_reserv=1
    for j in range(len(dfMuSigma)):
        if (j!=i):
            if (dfMuSigma['mytype'][j]=='norm'):
                init_reserv*=dfMuSigma['mu'][j]
            else:
                init_reserv*=np.exp(dfMuSigma['mu'][j]+0.5*dfMuSigma['sigma'][j]**2)

    if (dfMuSigma['mytype'][i]=='norm'):
        tek_mas=stats.norm.rvs(loc=dfMuSigma.iloc[i,0], scale=dfMuSigma.iloc[i,1], size=N)
        lower=init_reserv*calc_cte_norm(0.1,dfMuSigma.iloc[i,0], dfMuSigma.iloc[i,1],0)
        upper=init_reserv*calc_cte_norm(0.1,dfMuSigma.iloc[i,0], dfMuSigma.iloc[i,1],1)
        dfOTV=dfOTV.append({'upper':upper,'lower':lower}, ignore_index=True)
    else:
        tek_mas=stats.norm.rvs(loc=dfMuSigma.iloc[i,0], scale=dfMuSigma.iloc[i,1], size=N)
        tek_mas=np.exp(tek_mas)
        lower=init_reserv*calc_cte_lognorm(0.1,dfMuSigma.iloc[i,0], dfMuSigma.iloc[i,1],0)
        upper=init_reserv*calc_cte_lognorm(0.1,dfMuSigma.iloc[i,0], dfMuSigma.iloc[i,1],1)
        dfOTV=dfOTV.append({'upper':upper,'lower':lower}, ignore_index=True)
    tek_mas*=init_reserv
    plt.hist(tek_mas/1000,bins=100,density=True,label=i)
plt.legend()
plt.show()
```



Now we can see the dependance of the distribution of the reserves from each of the paramers. From the table we also see that the parameter with the largest difference between upper and lower CVaR is porosity (k), so we need to investigate this parameter more.

```
dfOTV['diff']=dfOTV['upper']-dfOTV['lower']
dfOTV.index=df1.index
dfOTV
```

	upper	lower	diff
k	1.705004e+06	153499.018865	1.551505e+06
h	8.448852e+05	398820.452697	4.460648e+05
so	8.448852e+05	398820.452697	4.460648e+05
s	6.588780e+05	584827.694788	7.405028e+04

4. Conclusions

We introduced the concepts of $POE, VaR, bPOE, CVaR$ and then explained why using $CVaR$ is preferable in the analysis of oil reserves. Moreover, utilizing $\overline{CVaR}$ and $\underline{CVaR}$, we suggested a method that enables the explorer to detect which parameter affects the reserves most of all and hence conduct extended measurements of this very parameter. We demonstrated this method on a numerical example, and found out that, according to given data, porosity affects the reserves more than other parameters. All calculations were conducted in Jupiter Notebook, and all of them are given.

References

1. A. Mafusalov, S. Uryasev. «Buffered probability of exceedance: mathematical properties and optimization». *Society for Industrial and Applied Mathematics*, 2018, Vol. 28, pp. 1077–1103
2. Justin R. Davis, Stan Uryasev. «Analysis of tropical storm damage using buffered probability of exceedance». Springer Science+Business Media Dordrecht, 2016