Estudo de Caso 3: Planejamento e Análise de Experimentos

Matheus Marzochi, Mayra Mota, Rafael Ramos e Victor Magalhães

11 de Novembro de 2019

Resumo

O objetivo deste estudo de caso é investigar como modificações de hiperparâmetros de um algoritmo de otimização baseado em evolução diferencial influenciam em seu desempenho, em diferentes cenários de execução. Os algoritmose serão testados partir de funções de Rosenbrook, de dimensões entre 2 e 150.

Papéis Desempenhados

A divisão de tarefas no grupo segue a descrição da *Declaração de Políticas de Equipe*. Estando aqui organizada da seguinte forma:

- Matheus: Verificador
- Mayra: Monitora
- Rafael: Coordenador
- Victor: Revisor

Planejamento do Experimento

Descrição do Problema

O problema analisado tem por objetivo comparar o desempenho de duas configurações de um algoritmo de otimização baseada em evolução diferencial, na resolução do problema de ótimo em funções de Rosenbrook. As funções podem possuir dimensões que variam entre 2 e 150. A hipótese nula é a de que o desempenho dos dois algoritmos permanece o mesma independente da configuração, e a hipótese que está sendo testada é a de que existe uma diferenção de desempenho entre elas.

Para avaliar o desempenho de cada um dos algoritmos, seus desempenhos sob diferentes dimensões da função de Rosenbrook são levados em consideração, a partir de testes pareados. Testes pareados constituem uma parte de testes blocados, onde distribuições são comparadas caso a caso, agrupadas em blocos, com o objetivo de diminuir efeitos que não estejam relacionados com os parâmetros em teste. O valor de performance y para cara algoritmo i em cada instância j é dado a partir da fórmula (Montgomery and Runger 2012):

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij}$$

onde μ corresponde à média geral de todas as amostras, τ_i corresponde o efeito do i-ésimo algoritmo, β_j representa , e ϵ_{ij} representa um erro randômico com média nula e independentemente distribuídos, de variância σ^2.

O que queremos testar é a equivalência dos parâmetros sob teste para cada algoritmo, ou seja, y_i. Seja y_i definido como:

$$y_i = \sum_{j=1}^{n} y_{ij}\overline{y_i} = \frac{y_i}{n}$$

onde n equivale ao total de instâncias (blocos). A hipótese nula é:

$$H_0 : y_1 = y_2 = ... = y_a$$

onde a equivale ao total de algoritmos sendo comparados, o que no problema em questão são 2. Como o que difere cada parâmetro y_i é o valor da influência do algoritmo i, dizer que a hipótese nula é que os parâmetros y_i são iguais equivale a dizer que

$$H_0 : \tau_1 = \tau_2 = ... = \tau_a = 0$$

o que indica não haver influência do algoritmo nos parâmetros das execuções, e a definição do mesmo passa a ser portanto apenas a média global acrescida de um erro ϵ_{ij}. Isto equivale a dizer que todas as observações foram retiradas de uma distribuição normal com média μ e variância σ^2. A hipótese em teste, é:

$$H_1 : \tau_i = 0$$

para pelo menos algum valor de i.

Para este trabalho, deseja-se saber se existe alguma diferença no desempenho médio do algoritmo quando carregado com diferentes configurações. O parâmetro utilizado foi baseado na diferença do desempenho médio das configurações 1 e 2, por tratar-se de uma análise pareada. Se as duas configurações apresentarem o mesmo desempenho, a diferença das médias de cada população amostrada será zero. Se uma configuração tiver o desempenho superior, o valor não será nulo. Desta forma, sendo $\mu = \mu_1 - \mu_2$, o teste deve possuir as seguintes hipóteses:

$$\begin{cases} H_0 : \mu = 0 \\ H_1 : \mu \neq 0 \end{cases}$$

Execução do Experimento

Como deseja-se avaliar o desempenho das duas configurações de algoritmos sob diferentes dimensões de problema, serão realizados testes pareados. Cada observação do teste corresponde a uma dimensão do problema de Rosenbrook. Considerando as métricas para o teste estabelecidas anteriormente, o número de instâncias para o teste pareado deverá ser, no mínimo, 34 amostras, como o cálculo a seguir demonstra.

```
result <- power.t.test(delta=0.5,
             sig.level=0.05,
             power=0.8,
             type='paired',
             alternative='two.sided')
print(result)
```

```
##
##      Paired t test power calculation
##
##               n = 33.3672
##           delta = 0.5
##              sd = 1
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs
```

Coleta dos Dados

Como, a princípio, não podemos assumir normalidade dos dados coletados, para saber a quantidade de instâncias a ser utilizada, fizemos o cálculo considerando o uso do teste de Wilcoxon (Lowry 2008), utilizando a função `calc_instances` (Campelo and Takahashi 2018).

```
Ncalc <- calc_instances(
    power=0.8,
    d=0.5,
    sig.level=0.05,
    alternative.side='two.sided',
    test='wilcoxon',
    ncomparisons=1
)
print(Ncalc$ninstances)
```

```
## [1] 40
```

O que nos indica o uso de, no mínimo, 45 instâncias. A partir de uma rotina de coleta, geramos arquivos csv com os resultados das execuções dos algoritmos.

```
n_instancias <- 45

dimensions <- sample(seq(2, 150), n_instancias)

out_dim.conf1 = data.frame(matrix(ncol = 2, nrow = 0))
colnames(out_dim.conf1) <- c("dim", "best")
out_dim.conf2 = data.frame(matrix(ncol = 2, nrow = 0))
colnames(out_dim.conf2) <- c("dim", "best")

count.dim <- 1
for (d in dimensions){

    dim <- ceil(d)

    for (r in 1:30) {


        cat("\nBuilding dimension ", dim)

        fn <- function(X){
            if(!is.matrix(X)) X <- matrix(X, nrow = 1)
            Y <- apply(X, MARGIN = 1,
                       FUN = smoof::makeRosenbrockFunction(dimensions = dim))
            return(Y)
        }
        selpars <- list(name = "selection_standard")
        stopcrit <- list(names = "stop_maxeval", maxevals = 5000*dim, maxiter = 100*dim)
        probpars <- list(name = "fn", xmin = rep(-5, dim), xmax = rep(10, dim))
        popsize = 5 * dim

        ## Config 1
        recpars1 <- list(name = "recombination_arith")
        mutpars1 <- list(name = "mutation_rand", f = 4)
        ## Config 2
        recpars2 <- list(name = "recombination_bin", cr = 0.7)
```

```
        mutpars2 <- list(name = "mutation_best", f = 3)

        out <- ExpDE(mutpars = mutpars1,
                     recpars = recpars1,
                     popsize = popsize,
                     selpars = selpars,
                     stopcrit = stopcrit,
                     probpars = probpars,
                     showpars = list(show.iters = "dots", showevery = 20))
        de <- list("dim"=dim, "best"=out$Fbest, "repetition"=r)
        out_dim.conf1 <- rbind(out_dim.conf1,de, stringsAsFactors=FALSE)

        out <- ExpDE(mutpars = mutpars2,
                     recpars = recpars2,
                     popsize = popsize,
                     selpars = selpars,
                     stopcrit = stopcrit,
                     probpars = probpars,
                     showpars = list(show.iters = "dots", showevery = 20))
        de <- list("dim"=dim, "best"=out$Fbest, "repetition"=r)
        out_dim.conf2 <- rbind(out_dim.conf2,de, stringsAsFactors=FALSE)

    }

    count.dim <- count.dim + 1
}

write.csv(out_dim.conf1, 'conf_1.csv', row.names=FALSE)
write.csv(out_dim.conf2, 'conf_2.csv', row.names=FALSE)
```

Ao final, temos um conjunto com 45 instâncias correspondentes a ordens do algoritmo de Rosenbrook, com 30 amostras em cada um.

Análise Exploratória dos Dados

Avaliando as amostras contidas em cada instância, uma análise de normalidade foi realizada:

```
conf_1 <- read.csv(file='conf_1.csv', sep=',')
conf_2<- read.csv(file='conf_2.csv', sep=',')

dataConf1 <- matrix(as.numeric(unlist(conf_1[,2])),nrow=nrow(conf_1))
dataConf2 <- matrix(as.numeric(unlist(conf_2[,2])),nrow=nrow(conf_2))

difs <- c()
all_conf_1 <- c()
all_conf_2 <- c()
for (i  in seq(1, nrow(dataConf1), 10)) {

    samples1 <- as.vector(dataConf1[i:(i+9),])
    samples2 <- as.vector(dataConf2[i:(i+9),])
    all_conf_1 <- c(all_conf_1, mean(samples1))
    all_conf_2 <- c(all_conf_2, mean(samples2))
    dif <- mean(samples1) - mean(samples2)
    difs <- c(difs, dif)
```

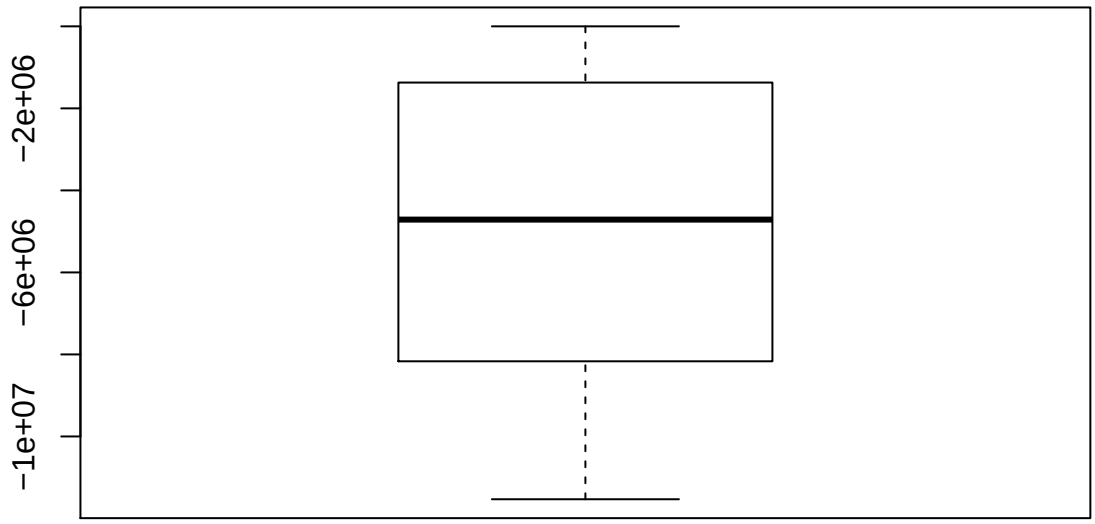

Figure 1: Box plot para as diferenças das médias.

```
}
```

```
##
##  Shapiro-Wilk normality test
##
## data:  difs
## W = 0.92502, p-value = 0.006281
```

De acordo com o teste de Shapiro-Wilk, o p-valor para a diferença foi de 0.006, abaixo da incerteza de 0.05, o que é um indício de que a distribuição dos dados não é normal.

O boxplot para as diferenças, como pode ser visto a seguir, mostra que existe uma assimetria nos dados, o que corrobora com a hipótese de não-normalidade. A partir da análise do diagrama, nota-se que os valores são negativos. Assim, essa disposição pode levar à conclusão da existência de uma diferença de desempenho entre as duas configurações. Entretanto, não é suficiente para inferir com confiança sobre a população.

Como visto, não se consegue negar a hipótese nula de que os dados vieram de uma distribuição normal, o que indica grande possibilidade de que os dados sejam normais. Pode-se, portanto, realizar um teste pareado com as médias dos dados obtidos a cada instância, usando teste de Wilcoxon (Lowry 2008).

```
result <- wilcox.test(difs, alternative='two.sided', conf.level=0.95)

print(result)
```

```
##
```

```
##  Wilcoxon signed rank test
##
## data:  difs
## V = 0, p-value = 5.684e-14
## alternative hypothesis: true location is not equal to 0
```

Os resultados obtidos pelo teste, portanto, foram:

- Graus de Liberdade = 44
- p-valor = 5.684 e-14

Teste de Hipótese

Discussão para Melhoria do Experimento

Conclusões

Referências

Campelo, Felipe, and Fernanda Takahashi. 2018. "Sample Size Estimation for Power and Accuracy in the Experimental Comparison of Algorithms."

Lowry, Richard. 2008. *Concepts & Applications of Inferential Statistics*.

Montgomery, Douglas, and George Runger. 2012. *Estatística Aplicada E Probabilidade Para Engenheiros*. LTC.