Method Of Moments

Josh Levine	Harsh Patel	Jaini Patel	Yifan Liao	Aayush Shah
jl2108	hkp49	jp1891	yl1463	avs93

What is Method of Moments?

In statistics, the method of moments is a method of estimation of population parameters such as mean, variance, median, etc. (which need not be moments), by equating sample moments with theoretical moments and then solving those equations for the quantities to be estimated.
Here are some definitions of theoretical moments:

1) $E(X^k)$ is the k^{th} (theoretical) moment of the distribution (about the origin), for $k = 1, 2, \ldots$
2) $E\left[(X-\mu)^k\right]$ is the k^{th} (theoretical) moment of the distribution (about the mean), for $k = 1, 2, \ldots$
3) $M_k = \frac{1}{n}\sum\limits_{i=1}^{n} X_i^k$ is the k^{th} sample moment, for $k = 1, 2, \ldots$
4) $M_k^* = \frac{1}{n}\sum\limits_{i=1}^{n}(X_i - \bar{X})^k$ is the k^{th} sample moment about the mean, for $k = 1, 2, \ldots$

The basic steps to find these moments are:

1) Equate the first sample moment about the origin $M_1 = \frac{1}{n}\sum\limits_{i=1}^{n} X_i = \bar{X}$ to the first theoretical moment $E(X)$
2) Equate the second sample moment about the origin $M_2 = \frac{1}{n}\sum\limits_{i=1}^{n} X_i^2$ to the second theoretical moment $E(X^2)$
3) Continue equating sample moments about the origin, M_k, with the corresponding theoretical moments $E(X^k),\ k = 3, 4, \ldots$ until you have as many equations as you have parameters.
4) Solve for the parameters.

The resulting values are called method of moments estimators.

The table shows the estimated parameters values found using method of moments estimator for a given distribution. The estimated parameters for distributions having two parameters, can be obtained by solving the equations obtained by solving the theoretical mean and the theoretical variance with the sample mean and sample variance, respectively and these estimated parameters are further determined as first and second moment about the sample mean.

Distribution	Actual Parameter 1	Actual Parameter 2	Estimated Parameter 1	Esitmated Parameter 2
Point Mass at a	a	0	$\hat{a}$ = $\hat{\mu}$	**N/A**
Bernoulli (p)	p	$p(1-p)$	$\hat{p}$ = $\hat{\mu}$	**N/A**
Binomial(n, p)	np	$np(1-p)$	$\hat{n}$ = $\hat{\mu}^2/(\hat{\mu}-\hat{\sigma}^2)$	$\hat{p}$ = $1-(\hat{\sigma}^2/\hat{\mu})$
Geometric(p)	$1/p$	$(1-p)/p^2$	$\hat{p}$ = $1/\hat{\mu}$	**N/A**
Poisson(λ)	λ	λ	$\hat{\lambda}$ = $\hat{\mu}$	**N/A**
Uniform(a, b)	$(a+b)/2$	$(b-a)^2/12$	$\hat{a}$ = $\hat{\mu}-\sqrt{3\hat{\sigma}^2}$	$\hat{b}$ = $\hat{\mu}+\sqrt{3\hat{\sigma}^2}$
Normal(μ, σ^2)	μ	σ^2	$\hat{\mu}_0$ = $\hat{\mu}$	$\hat{\sigma}_0{}^2$ = $\hat{\sigma}^2$
Exponential(β)	β	β^2	$\hat{\beta}$ = $1/\hat{\mu}$	**N/A**
Gamma(α, β)	$\alpha\beta$	$\alpha\beta^2$	$\hat{\alpha}$ = $\hat{\mu}^2/\hat{\sigma}^2$	$\hat{\beta}$ = $\hat{\sigma}^2/\hat{\mu}$
Beta(α, β)	$\alpha/\alpha+\beta$	$\alpha\beta/(\alpha+\beta)^2(\alpha+\beta+1)$	$\hat{\alpha}$ = $\hat{\mu}[\hat{\mu}(1-\hat{\mu})/\hat{\sigma}^2-1]$	$\hat{\beta}$ = $(1-\hat{\mu})[\hat{\mu}(1-\hat{\mu})/\hat{\sigma}^2-1]$
t_v	$0(ifv \geq 1)$	$v/(v-2)(ifv>2)$	$\hat{v}$ = $2t/(1-t)$	
Chi-Square(p)	p	$2p$	$\hat{p}$ = $\hat{\mu}$	**N/A**
Multinomial(n, p)	np			
Multivariate Normal($\mu, \sum$)	μ	$\sum$		

Method of Moments Function used to call each distribution

Here, is the code that we have implemented in order to calculate first and second moment for various distributions, on a sample size of 1000. firstMoment and secondMoment are the functions to compute first and second moment for the given distribution on a sample space. methodOfMoment is the function that computes the estimated parameters with the help of given parameters

```
# Compute 1st moment for the sample around origin to get Sample_Mean
firstMoment <- function(vec){
  sum = sum(vec, na.rm=TRUE)
  return (sum/length(vec))
}

# Compute 2nd moment for the sample around the sample mean to get Sample_Variance
secondMoment <- function(vec){
  mu = firstMoment(vec)
  square <- sum((vec - mu)^2, na.rm=TRUE)
  return (square/length(vec))
}

methodOfMoment <- function(n, distribution){

  cat(paste("The distribution is:", distribution ,"\n" ))

  if(distribution == "point"){
    vec = n
    a_hat = point(vec)
    print(paste0("Estimated parameter:", "a_hat = ", a_hat))

  }

  if( distribution == "bernoulli" ){
    p = 0.4
    vec = rbern(n, p)
    p_hat = bernoulli(vec)
    print(paste0("Population Parameter:   ", "p = ", p))
    print(paste0("Estimated parameter:", "p_hat = ", round(p_hat,3) ))
  }

  if( distribution == "binomial" ){
    n_ = 50
    p = 0.6
    vec = rbinom(n, n_, p)
    l = binomial(vec)
    print(paste0("Population Parameters:    ", "n_ = ", n_, "             p = ", p))
    print(paste0("Estimated parameters: ", "n_hat = ",round(l$n_hat,3) , "      p_hat = ", round
(l$p_hat,3)) )
  }
```

```
if( distribution == "geometric" ){
    p = 0.8
    vec = rgeom(n, p)
    p_hat = geometric(vec)
    print(paste0("Population Parameter:   ", "p = ", p))
    print(paste0("Estimated parameter:", "p_hat = ",round(p_hat,3) ))
  }

 if( distribution == "poisson" ){
    lambda = 0.2
    vec = rpois(n, lambda)
    lambda_hat = poisson(vec)
    print(paste0("Population Parameter:   ", "lambda = ", lambda))
    print(paste0("Estimated parameter:", "lambda_hat = ",round(lambda_hat,3) ))
  }

 if( distribution == "uniform" ){
    a = 10
    b = 30
    vec = runif(n, a, b)
    l = uniform(vec)
    print(paste0("Population Parameters:    ", "a = ", a, "               b = ", b))
    print(paste0("Estimated parameters: ", "a_hat = ",round(l$a_hat,3), "      b_hat = ", round
(l$b_hat,3) ) )
  }

  if( distribution == "normal" ){
    mu = 20
    sd = 16
    vec = rnorm(n, mu, sd)
    l = normal(vec)
    print(paste0("Population Parameters:    ", "mu = ", mu, "              sd = ", sd))
    print(paste0("Estimated parameters: ", "mu_hat = ",round(l$mu_hat,3), "      sd_hat = ", rou
nd(l$sd_hat,3) ) )
  }
if( distribution == "exponential" ){
    beta = 20
    vec = rexp(n, beta)
    beta_hat = exponential(vec)
    print(paste0("Population Parameter:   ", "beta = ", beta))
    print(paste0("Estimated parameter:", "beta_hat = ",round(beta_hat,3) ))
  }

  if( distribution == "gamma" ){
    alpha = 3
    beta = 0.5
    vec = rgamma(n, alpha, scale = beta)
    l = gamma(vec)
    print(paste0("Population Parameters:    ", "alpha = ", alpha, "               beta = ", bet
a))
    print(paste0("Estimated parameters: ", "alpha_hat = ",round(l$alpha_hat,3), "      beta_hat
= ", round(l$beta_hat,3) ) )
  }
```

```
if( distribution == "beta" ){
    alpha = 4
    beta = 10
    vec = rbeta(n, alpha, beta)
    l = beta(vec)
    print(paste0("Population Parameters:    ", "alpha = ", alpha, "               beta = ", bet
a))
    print(paste0("Estimated parameters: ", "alpha_hat = ",round(l$alpha_hat,3), "      beta_hat
= ", round(l$beta_hat,3) ) )
  }

 if( distribution == "t" ){
    v = 10
    vec = rt(n, df = v)
    v_hat = t(vec)
    print(paste0("Population Parameter:   ", "v = ", v))
    print(paste0("Estimated parameter:", "v_hat = ",round(v_hat,3) ))
  }

if( distribution == "chi_squared" ){
    p = 5
    vec = rchisq(n, p)
    p_hat = chi_squared(vec)
    print(paste0("Population Parameter:   ", "p = ", p))
    print(paste0("Estimated parameter:", "p_hat = ",round(p_hat,3) ))
  }

  if(distribution == "multinomial"){
    p = c(0.15,0.05,0.4,0.1,0.3)
    input_data = rmultinom(10000,size=5,p)
    a = nrow(input_data)
    print(paste0("Population parameters: ", "p  = ", p, " a = ", a))
    print("Multinomial distribution has following estimated parameters: ")
    sample_data = input_data
    estimator <- multinomial(sample_data)
    print(estimator)
  }

  if(distribution == "multivariate_normal"){
    vari = c(10,3,3,2)
    sigma = matrix(vari,2,2)
    input_data = mvrnorm(n = 1000, rep(0, 2), sigma)
    print("Population parameter: ")
    print(vari)
    print("Multinormal distribution has following estimated parameters")
    sample_data = input_data
    estimator <- multivariate_normal(sample_data)
    print(estimator)
  }

  cat("\n")
```

Below are the functions to compute the outputs and the outputs for each distributions on a sample space of size 1000.

Point Mass

```
# Point Distribution
point <- function(vec){
  # Estimating the parameters
  a_hat = firstMoment(vec)
  return(a_hat)
}

methodOfMoment(1000, "point")
```

```
## The distribution is: point
## [1] "Estimated parameter:a_hat = 1000"
```

Bernoulli

```
bernoulli <- function(vec){
  p_hat = firstMoment(vec)
  return(p_hat)
}

methodOfMoment(1000, "bernoulli")
```

```
## The distribution is: bernoulli
## [1] "Population Parameter:   p = 0.4"
## [1] "Estimated parameter: p_hat = 0.401"
```

Binomial

```
binomial <- function(vec){
  mu = firstMoment(vec)
  var = secondMoment(vec)

  p_hat = 1 - (var/mu)
  n_hat = mu/p_hat

  list(n_hat = n_hat, p_hat = p_hat)
}

methodOfMoment(1000, "binomial")
```

```
## The distribution is: binomial
## [1] "Population Parameters:    n_ = 50             p = 0.6"
## [1] "Estimated parameters: n_hat = 48.678      p_hat = 0.618"
```

Geometric

```r
geometric <- function(vec){
  p_hat = 1/firstMoment(vec) - 1
  return(p_hat)
}

methodOfMoment(1000, "geometric")
```

```
## The distribution is: geometric
## [1] "Population Parameter:   p = 0.8"
## [1] "Estimated parameter:p_hat = 3.444"
```

Poisson

```r
poisson <- function(vec){
  lambda_hat = firstMoment(vec)
  return(lambda_hat)
}

methodOfMoment(1000, "poisson")
```

```
## The distribution is: poisson
## [1] "Population Parameter:   lambda = 0.2"
## [1] "Estimated parameter:lambda_hat = 0.201"
```

Uniform

```r
uniform <- function(vec){
  mu = firstMoment(vec)
  var = secondMoment(vec)
  hold = var*sqrt(3)

  a_hat = mu - hold
  b_hat = mu + hold

  list(a_hat = a_hat, b_hat = b_hat)
}

methodOfMoment(1000, "uniform")
```

```
## The distribution is: uniform
## [1] "Population Parameters:    a = 10               b = 30"
## [1] "Estimated parameters: a_hat = -35.142      b_hat = 75.216"
```

Normal

```
normal <- function(vec){
  mu_hat = firstMoment(vec)
  sd_hat = sqrt(secondMoment(vec))

 list(mu_hat = mu_hat, sd_hat = sd_hat)
}

methodOfMoment(1000, "normal")
```

```
## The distribution is: normal
## [1] "Population Parameters:    mu = 20              sd = 16"
## [1] "Estimated parameters: mu_hat = 19.479      sd_hat = 15.303"
```

Exponential

```
exponential <- function(vec){
  beta_hat = 1/firstMoment(vec)
  return(beta_hat)
}

methodOfMoment(1000, "exponential")
```

```
## The distribution is: exponential
## [1] "Population Parameter:   beta = 20"
## [1] "Estimated parameter:beta_hat = 20.021"
```

Gamma

```
gamma <- function(vec){
  mu = firstMoment(vec)
  var = secondMoment(vec)

  alpha_hat = (mu^2)/(var^2)
  beta_hat = (var^2)/mu

  list(alpha_hat = alpha_hat, beta_hat = beta_hat)
}

methodOfMoment(1000, "gamma")
```

```
## The distribution is: gamma
## [1] "Population Parameters:    alpha = 3               beta = 0.5"
## [1] "Estimated parameters: alpha_hat = 4.133      beta_hat = 0.373"
```

Beta

```
beta <- function(vec){
  mu = firstMoment(vec)
  var = secondMoment(vec)
  hold = (mu*(1-mu)/var) - 1

  alpha_hat = mu * hold
  beta_hat = (1-mu) * hold

  list(alpha_hat = alpha_hat, beta_hat = beta_hat)
}

methodOfMoment(1000, "beta")
```

```
## The distribution is: beta
## [1] "Population Parameters:    alpha = 4               beta = 10"
## [1] "Estimated parameters: alpha_hat = 4.22      beta_hat = 10.608"
```

t

```
t <- function(vec){
  mu = firstMoment(vec)
  if(mu != 0){
    v_hat = 2*mu/(mu-1)
    return(v_hat)
    }
  else{
    return(0)
  }
}

methodOfMoment(1000, "t")
```

```
## The distribution is: t
## [1] "Population Parameter:   v = 10"
## [1] "Estimated parameter:v_hat = 0.069"
```

Chi Squared

```
chi_squared <- function(vec){
  p_hat = firstMoment(vec)
  return(p_hat)
}

methodOfMoment(1000, "chi_squared")
```

```
## The distribution is: chi_squared
## [1] "Population Parameter:   p = 5"
## [1] "Estimated parameter:p_hat = 4.982"
```

Multinomial

```
multinomial <- function(vec){
  a = nrow(vec)
  p = c(0,0,0,0,0)
  for(i in 1:a)
    p[i]<-1-((var(vec[i,]))/mean(vec[i,]))
  n = sum(rowMeans(vec))/sum(p[1:a])
  return (n)
}

methodOfMoment(1000, "multinomial")
```

```
## The distribution is: multinomial
## [1] "Population parameters: p  = 0.15 a = 5"
## [2] "Population parameters: p  = 0.05 a = 5"
## [3] "Population parameters: p  = 0.4 a = 5"
## [4] "Population parameters: p  = 0.1 a = 5"
## [5] "Population parameters: p  = 0.3 a = 5"
## [1] "Multinomial distribution has following estimated parameters: "
## [1] 4.988744
```

Multivariate Normal

```
multivariate_normal <- function(vec){
  mu_hat = colMeans(vec)
  summation = var(vec)
  print("For Multinormal Distribution the parameter mu_hat is")
  print(mu_hat)
  print("The parameter summation is ")
  print(summation)
}

methodOfMoment(1000, "multivariate_normal")
```

```
## The distribution is: multivariate_normal
## [1] "Population parameter: "
## [1] 10  3  3  2
## [1] "Multinormal distribution has following estimated parameters"
## [1] "For Multinormal Distribution the parameter mu_hat is"
## [1] 0.06266782 0.02084492
## [1] "The parameter summation is "
##          [,1]     [,2]
## [1,] 9.655650 2.765244
## [2,] 2.765244 1.913141
##          [,1]     [,2]
## [1,] 9.655650 2.765244
## [2,] 2.765244 1.913141
```

Conclusion:

Distribution	Actual Parameter 1	Actual Parameter 2	Estimated Parameter 1	Esitmated Parameter 2
Point Mass at a	a = 1000	N/A	$\hat{a}$ = 1000	N/A
Bernoulli (p)	p = 0.4	N/A	$\hat{p}$ = 0.401	N/A
Binomial(n, p)	n = 50	p = 0.6	$\hat{n}$ = 48.678	$\hat{p}$ = 0.618
Geometric(p)	p = 0.8	N/A	$\hat{p}$ = 3.444	N/A
Poisson(λ)	λ = 0.2	N/A	$\hat{\lambda}$ = 0.201	N/A
Uniform(a, b)	a = 10	b = 30	$\hat{a}$ = -35.142	$\hat{b}$ = 75.216
Normal(μ, σ^2)	μ = 20	σ = 16	$\hat{\mu}$ = 19.479	$\hat{\sigma}$ = 15.303
Exponential(β)	β = 20	N/A	$\hat{\beta}$ = 20.021	N/A
Gamma(α, β)	α = 3	β = 0.5	$\hat{\alpha}$ = 4.133	$\hat{\beta}$ = 0.373
Beta(α, β)	α = 4	β = 10	$\hat{\alpha}$ = 4.22	$\hat{\beta}$ = 10.608
t_v	v = 10	N/A	$\hat{v}$ = 0.069	N/A
Chi-Square(p)	p = 5	N/A	$\hat{p}$ = 4.982	N/A

From our output results, we can say that the estimated values using the method of moment estimator is quite consistent for most of the probability distributions with at most two parameters (when the sample size is 1000). Our estimator accurately estimated the parameter values with an error of ~0.5, which is not a small value. We can say, that this estimating method will provide good estimates, since the empirical distribution converges in some sense to the probability distribution, making the values of parameters almost equal. But, better methods like maximum likelihood can be used in order to get more accurate results.

References:

Reference 1 (https://online.stat.psu.edu/stat415/lesson/1/1.4#fullScreen)