A Fast Algorithm for Computing Product Moments of Multivariate Normal Random Variables

Raymond Kan[a,*], Jiening Pan[b]

[a]Rotman School of Management, University of Toronto, 105 St. George Street, Ontario, Canada, M5S 3E6
[b]Nankai University, 38 Tongyan Road, Jinnan District, Tianjin, Tianjin 300350, China

Abstract

We provide a simple identity that decomposes a product moment of multivariate normal random variables as a sum of various products of univariate moments of one of the random variables and multivariate moments of the other random variables. The new identity allows for much faster computation of the product moments of multivariate normal random variables than existing methods.

Keywords: Moments of product, Multivariate normal distribution.
2020 MSC: Primary 60E10, Secondary 62H10

1. Introduction

Let $z = [z_1, \dots, z_n]^\top \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a multivariate normal random vector with expected value $\boldsymbol{\mu} = [\mu_1, \dots, \mu_n]^\top$ and covariance matrix $\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma} = (\sigma_{ij})$ is a positive semidefinite matrix. For $\boldsymbol{s} = (s_1, \dots, s_n)$, where s_i's are nonnegative integers, we are interested in obtaining computationally efficient expressions for the expectation of a product of the elements of z,

$$\mu'_s \equiv \mathrm{E}[z_1^{s_1} z_2^{s_2} \cdots z_n^{s_n}].$$

Explicit and recursive expressions of this product moment are available in the statistics literature. When $s_1 = \dots = s_n = 1$ and $\boldsymbol{\mu} = \boldsymbol{0}$, the explicit formula for μ'_s is available since Isserlis [4]. In physics literature, Isserlis's formula is often written as the hafnian of Σ and it is known as the Wick's formula. However, for even $s = s_1 + \cdots + s_n$, this formula requires summing up $(s-1)!! = 1 \times 3 \times \cdots \times (s-1)$ terms of product of $s/2$ elements of the Σ matrix. Even for moderately large s, the number of calculations is astronomical. For example, if one wishes to calculate $\mathrm{E}[z_1 \cdots z_{20}]$, then one would need to sum up 19!! = 654,729,075 number of terms to obtain the answer, which is clearly impractical. For general $\boldsymbol{s}$ and $\boldsymbol{\mu} = \boldsymbol{0}$, the explicit formula of μ'_s is available from [2] and [6]. Even for moderately large s, this explicit formula requires summing up a large number of terms.

For the noncentral case, i.e., $\boldsymbol{\mu} \neq \boldsymbol{0}$, [7] recently presented an explicit formula of μ'_s in terms of the elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Naturally, this formula has even more terms than the formula for the central case because the former also involves the elements of $\boldsymbol{\mu}$. For $\mathrm{E}[(z_1z_2z_3z_4z_5)^4]$, the explicit formula only requires summing up 99,450 terms, which is quite manageable, but the number of terms goes up to 63,637,506 for $\mathrm{E}[(z_1 \cdots z_{10})^2]$ and 23,758,664,096 for $\mathrm{E}[z_1 \cdots z_{20}]$. Even for the simple case of $\mathrm{E}[(z_1z_2z_3)^{100}]$, the explicit formula would require summing up 3,321,449,001 terms. As a result, this explicit formula is not ideal for the numerical computation of μ'_s.

Using a recurrence relation on multivariate Hermite polynomials (see [8]), [9] presents an $(n+1)$-term recursive formula for computing μ'_s. This recursive method is quite efficient, but a large number of terms is still required to obtain μ'_s. For example, computing $\mathrm{E}[(z_1z_2z_3)^{100}]$ using the recursive algorithm would require summing up 4,090,594 terms because we need to compute $\mathrm{E}[z_1^i z_2^j z_3^k]$ for $0 \le i \le 100$, $0 \le j \le 100$, and $0 \le k \le 100$.

Based on a formula that relates the moment of a product of random variables to moments of various sums of the random variables, [5] provides an alternative approach for computing μ'_s. In many cases, this new formula can

*Corresponding author. Email address: Raymond.Kan@rotman.utoronto.ca

provide a substantial improvement over the explicit formula. For example, it requires summing up only 324,764 terms for computing $\mathrm{E}[(z_1 \cdots z_{10})^2]$. However, it still requires summing up 77,787,650 terms for computing $\mathrm{E}[(z_1 z_2 z_3)^{100}]$, which is extremely time-consuming.

In this paper, we develop a new formula that provides a substantial improvement on the speed for computing μ'_s. Our starting point is the explicit formula for μ'_s from [7]. However, instead of expressing μ'_s as a sum of various products of elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we decompose their formula into sum of products of univariate moments of z_1 and multivariate moments of $(z_2, \cdots, z_n)$. In many cases, this new formula has far fewer terms than existing methods. For example, our new method can compute $\mathrm{E}[(z_1 z_2 z_3 z_4 z_5)^4]$ with 2,692 terms, $\mathrm{E}[(z_1 \cdots z_{10})^2]$ with 137,819 terms, and $\mathrm{E}[z_1 \cdots z_{20}]$ with 5,505,005 terms. For the case of $\mathrm{E}[(z_1 z_2 z_3)^{100}]$, our new formula requires summing up only 35,745 terms, which offers a significant improvement over existing methods.

The rest of the paper is organized as follows. Section 2 presents our new method for the $n = 2$ and $n = 3$ cases. This serves as the motivation for the development of our new method for the general case, which is presented in Section 3. Section 4 provides a further generalization of the results in Section 3. Section 5 concludes the paper.

2. Motivation

In this section, we first present our new algorithms for computing μ'_s for the cases of $n = 2$ and $n = 3$. This allows us to illustrate the advantages of the new algorithms over existing methods. The algorithms for the general case will be presented in Section 3.

When $n = 1$, we have $z \sim \mathcal{N}(\mu, \sigma^2)$, and we can obtain an explicit expression of $\mu'_s \equiv \mathrm{E}[z^s]$ by using

$$\mu'_s = \mathrm{E}[(z - \mu + \mu)^s] = \sum_{j=0}^{\lfloor s/2 \rfloor} \binom{s}{2j} \mathrm{E}[(z-\mu)^{2j}]\mu^{s-2j} = \sum_{j=0}^{\lfloor s/2 \rfloor} d_{s,j}\sigma^{2j}\mu^{s-2j}, \tag{1}$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x and

$$d_{s,j} = \frac{s!}{2^j j!(s-2j)!}. \tag{2}$$

From [1], we can also obtain μ'_s by using the following two-term recursive relation

$$\mu'_{s+1} = \mu\mu'_s + s\sigma^2\mu'_{s-1}, \tag{3}$$

with the initial conditions of $\mu'_0 = 1$ and $\mu'_1 = \mu$.

When $n = 2$, we have

$$\boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} \right).$$

[7] provides an explicit expression of $\mu'_{s_1,s_2} \equiv \mathrm{E}[z_1^{s_1} z_2^{s_2}]$ as

$$\mu'_{s_1,s_2} = \sum_{j=0}^{\min[s_1,s_2]} \sum_{p=0}^{\lfloor (s_1-j)/2 \rfloor} \sum_{q=0}^{\lfloor (s_2-j)/2 \rfloor} d_{(s_1,s_2),(p,j,q)} \sigma_{12}^j \sigma_{11}^p \sigma_{22}^q \mu_1^{s_1-j-2p} \mu_2^{s_2-j-2q}, \tag{4}$$

where[1]

$$d_{(s_1,s_2),(p,j,q)} = \frac{s_1! s_2!}{2^{p+q} j! p! q! (s_1-j-2p)!(s_2-j-2q)!}. \tag{5}$$

We now present a new method of computing μ'_{s_1,s_2}. This is obtained by recognizing $d_{(s_1,s_2),(p,j,q)}$ in (5) can be decomposed as

$$d_{(s_1,s_2),(p,j,q)} = \frac{s_1! s_2!}{j!(s_1-j)!(s_2-j)!} d_{s_1-j,p} d_{s_2-j,q},$$

[1]Our $d_{(s_1,s_2),(p,j,q)}$ is actually defined as $d_{s_1,s_2,j,p,q}$ in [7]. We adopt a slightly different notation here for the convenience of extending the results to the general case of n.

where $d_{s,j}$ is defined in (2). This allows us to write

$$\frac{\mu'_{s_1,s_2}}{s_1!s_2!} = \sum_{j=0}^{\min[s_1,s_2]} \frac{\sigma_{12}^j}{j!} \frac{E[z_1^{s_1-j}]}{(s_1-j)!} \frac{E[z_2^{s_2-j}]}{(s_2-j)!} = \sum_{j=0}^{\min[s_1,s_2]} \frac{\sigma_{12}^j}{j!} \frac{\mu'_{s_1-j,0}}{(s_1-j)!} \frac{\mu'_{0,s_2-j}}{(s_2-j)!},$$

or equivalently

$$\tilde{\mu}'_{s_1,s_2} = \sum_{j=0}^{\min[s_1,s_2]} \frac{\sigma_{12}^j}{j!} \tilde{\mu}'_{s_1-j,0} \tilde{\mu}'_{0,s_2-j}, \tag{6}$$

where $\tilde{\mu}'_{i,j} = \mu'_{i,j}/(i!j!)$. This identity allows us to compute the bivariate moments of (z_1, z_2) by using the univariate moments of z_1 and z_2, and these univariate moments can be obtained by using the recursive relation from (3).

In addition to the explicit expression, there is the following three-term recursive relation for μ'_{s_1,s_2}, which is given by [9]

$$\mu'_{s_1+1,s_2} = \mu_1 \mu'_{s_1,s_2} + s_1 \sigma_{11} \mu'_{s_1-1,s_2} + s_2 \sigma_{12} \mu'_{s_1,s_2-1} \tag{7}$$

for $s_1 \geq 0$ and $s_2 \geq 0$, with the initial conditions of $\mu'_{0,0} = 1$, $\mu'_{1,0} = \mu_1$, and $\mu'_{0,1} = \mu_2$.

Using an identity that relates the product moment to moment of sums, [5] provides yet another expression for computing μ'_{s_1,s_2}, which is given by

$$\mu'_{s_1,s_2} = \sum_{\nu_1=0}^{s_1} \sum_{\nu_2=0}^{s_2} \sum_{r=0}^{\lfloor s/2 \rfloor} (-1)^{\nu_1+\nu_2} \binom{s_1}{\nu_1} \binom{s_2}{\nu_2} \frac{\left(\frac{\boldsymbol{h}_\nu^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{h}_\nu}{2}\right)^r (\boldsymbol{h}_\nu^{\mathrm{T}} \boldsymbol{\mu})^{s-2r}}{r!(s-2r)!}, \tag{8}$$

where $s = s_1 + s_2$ and $\boldsymbol{h}_\nu = [s_1/2 - \nu_1,\ s_2/2 - \nu_2]^{\mathrm{T}}$.

For $n = 2$, the number of terms required by different methods for computing μ'_{s_1,s_2} are given in Table 1, assuming $s_1 \geq s_2$. For a fair comparison, when counting the number of terms for (6) in Table 1, we also add the number of terms that are needed to create the univariate moments of z_1 and z_2.

Table 1: Number of terms for computing μ'_{s_1,s_2}

Method	Number of Terms
Song and Lee (2015) (4)	
Odd s_2	$(s_2+1)(s_2+3)(4+3s_1-s_2)/24$
Even s_2	$(s_2+2)[12\lfloor (s_1+2)/2 \rfloor + (2+3s_1-s_2)s_2]/24$
New Algorithm (6)	$2s + s_2 - 3$
Willink (2005) (7)	$2(s-2) + 3s_1s_2$
Kan (2008) (8)	$\lfloor (s_1+1)(s_2+1)/2 \rfloor \lfloor s/2+1 \rfloor$

For comparison, Table 2 presents the number of terms for different methods under different combinations of (s_1, s_2). From Table 2, we can see that the explicit formula (4) has the fewest number of terms when $s_2 = 1$.[2] When s_1 and s_2 are both large, computing μ'_{s_1,s_2} using the recursive relation of (7) can be more efficient. For example, when $s_1 = s_2 = 100$, (4) requires summing up 88,451 terms, whereas the recursive relation (7) only requires summing up 30,396 terms. When $n = 2$, the formula based on [5] always involves the most number of terms and it is not advisable to use (8) for this case. Finally, our new expression in (6) requires summing up the fewest number of terms when $s_2 > 1$. The reduction of number of terms can be quite substantial. For example, (6) requires summing up only 497 terms for computing $\mu'_{100,100}$, which is vastly superior to all the other methods.

[2]In terms of computation speed, (6) still dominates (4) for $s_2 = 1$ because the latter requires the computation of $d_{(s_1,s_2),(p,j,q)}$ in each term.

Table 2: Number of terms for computing μ'_{s_1,s_2} under different combinations of (s_1, s_2)

(s_1, s_2)	Number of Terms (4)	(6)	(7)	(8)
(1,1)	2	2	3	4
(4,1)	5	8	18	15
(4,4)	19	17	60	60
(10,1)	11	20	48	66
(10,4)	46	29	144	216
(10,10)	146	47	336	660
(100,1)	101	200	498	5151
(100,4)	451	209	1404	13356
(100,10)	1766	227	3216	31080
(100,100)	88451	497	30396	515100

When $n = 3$, we have

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}, \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix}\right).$$

[7] presents an explicit expression of $\mu'_{s_1,s_2,s_3} \equiv \mathrm{E}[z_1^{s_1} z_2^{s_2} z_3^{s_3}]$ as

$$\mu'_{s_1,s_2,s_3} = \sum_{l \in \mathcal{L}_s} d_{s,l} \left(\prod_{i=1}^{3}\prod_{j=i}^{3} \sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{n} \mu_i^{L_{s,i}}\right), \tag{9}$$

where $s = (s_1, s_2, s_3)$, $l = (l_{11}, l_{12}, l_{13}, l_{22}, l_{23}, l_{33})$,

$$\begin{aligned} L_{s,1} &= s_1 - 2l_{11} - l_{12} - l_{13}, \\ L_{s,2} &= s_2 - 2l_{22} - l_{12} - l_{23}, \\ L_{s,3} &= s_3 - 2l_{33} - l_{13} - l_{23}, \\ d_{s,l} &= \frac{s_1!s_2!s_3!}{2^{l_{11}+l_{22}+l_{33}} l_{11}!l_{22}!l_{33}!l_{12}!l_{13}!l_{23}!L_{s,1}!L_{s,3}!L_{s,3}!}, \end{aligned} \tag{10}$$

and $\mathcal{L}_s$ is the set of nonnegative integers $l's$ such that $L_{s,i} \geq 0$ for $i = 1, 2, 3$. More explicitly, we can write $\sum_{l \in \mathcal{L}_s}$ as

$$\sum_{l_{12}=0}^{\min[s_1,s_2]} \sum_{l_{13}=0}^{\min[s_3,s_1-l_{12}]} \sum_{l_{23}=0}^{\min[s_2-l_{12},s_3-l_{13}]} \sum_{l_{11}=0}^{\lfloor (s_1-l_{12}-l_{13})/2 \rfloor} \sum_{l_{22}=0}^{\lfloor (s_2-l_{12}-l_{23})/2 \rfloor} \sum_{l_{33}=0}^{\lfloor (s_3-l_{13}-l_{23})/2 \rfloor}$$

Using (2) and (5), we can verify that $d_{s,l}$ in (10) can be decomposed as

$$d_{s,l} = \frac{s_1!s_2!s_3!}{(s_1 - l_{12} - l_{13})!(s_2 - l_{12})!(s_3 - l_{13})!l_{12}!l_{13}!} d_{s_1-l_{12}-l_{13},l_{11}} d_{(s_2-l_{12},s_3-l_{13}),(l_{22},l_{23},l_{33})}.$$

Then using (1) and (4), we obtain

$$\mathrm{E}[z_1^{s_1-l_{12}-l_{13}}] = \sum_{l_{11}=0}^{\lfloor (s_1-l_{12}-l_{13})/2 \rfloor} d_{s_1-l_{12}-l_{13},l_{11}} \sigma_{11}^{l_{11}} \mu_1^{s_1-l_{12}-l_{13}-2l_{11}},$$

$$\mathrm{E}[z_2^{s_2-l_{12}} z_3^{s_3-l_{13}}] = \sum_{l_{23}=0}^{\min[s_2-l_{12},s_3-l_{13}]} \sum_{l_{22}=0}^{\lfloor (s_2-l_{12}-l_{23})/2 \rfloor} \sum_{l_{33}=0}^{\lfloor (s_3-l_{13}-l_{23})/2 \rfloor} d_{(s_2-l_{12},s_3-l_{13}),(l_{22},l_{23},l_{33})} \sigma_{23}^{l_{23}} \sigma_{22}^{l_{22}} \sigma_{33}^{l_{33}} \mu_2^{s_2-l_{12}-l_{23}-2l_{22}} \mu_3^{s_3-l_{13}-l_{23}-2l_{33}},$$

and hence we can write

$$\frac{\mu'_{s_1,s_2,s_3}}{s_1!s_2!s_3!} = \sum_{l_{12}=0}^{\min[s_1,s_2]}\sum_{l_{13}=0}^{\min[s_3,s_1-l_{12}]} \frac{\sigma_{12}^{l_{12}}\sigma_{13}^{l_{13}}}{l_{12}!l_{13}!}\frac{\mathrm{E}[z_1^{s_1-l_{12}-l_{13}}]}{(s_1-l_{12}-l_{13})!}\frac{\mathrm{E}[z_2^{s_2-l_{12}}z_3^{s_3-l_{13}}]}{(s_2-l_{12})!(s_3-l_{13})!}.$$

Replacing l_{12} and l_{13} by i and j, we obtain our new formula for μ'_{s_1,s_2,s_3}:

$$\tilde{\mu}'_{s_1,s_2,s_3} = \sum_{i=0}^{\min[s_1,s_2]}\sum_{j=0}^{\min[s_3,s_1-i]} \frac{\sigma_{12}^{i}\sigma_{13}^{j}}{i!j!}\tilde{\mu}'_{s_1-i-j,0,0}\tilde{\mu}'_{0,s_2-i,s_3-j}, \tag{11}$$

where $\tilde{\mu}'_{i,j,k} = \mu'_{i,j,k}/(i!j!k!)$.

This new formula decomposes a trivariate moment of (z_1, z_2, z_3) into a sum of various products of univariate moments of z_1 and bivariate moments of (z_2, z_3). These univariate moments and bivariate moments can be easily obtained by using (3) and (7). If we apply (6) to decompose $\tilde{\mu}'_{0,s_2-i,s_3-j}$ in (11), we obtain another formula for $\tilde{\mu}'_{s_1,s_2,s_3}$ that is in terms of just the univariate moments of z_1, z_2, and z_3:

$$\tilde{\mu}'_{s_1,s_2,s_3} = \sum_{i=0}^{\min[s_1,s_2]}\sum_{j=0}^{\min[s_3,s_1-i]}\sum_{k=0}^{\min[s_2-i,s_3-j]} \frac{\sigma_{12}^{i}\sigma_{13}^{j}\sigma_{23}^{k}}{i!j!k!}\tilde{\mu}'_{s_1-i-j,0,0}\tilde{\mu}'_{0,s_2-i-k,0}\tilde{\mu}'_{0,0,s_3-j-k}. \tag{12}$$

As for the case of $n = 2$, there are two other methods for computing μ'_{s_1,s_2,s_3}. The recursive relation for computing μ'_{s_1,s_2,s_3} is given by [9]:

$$\mu'_{s_1+1,s_2,s_3} = \mu_1\mu'_{s_1,s_2,s_3} + s_1\sigma_{11}\mu'_{s_1-1,s_2,s_3} + s_2\sigma_{12}\mu'_{s_1,s_2-1,s_3} + s_3\sigma_{13}\mu'_{s_1,s_2,s_3-1} \tag{13}$$

for $s_1 \geq 0$, $s_2 \geq 0$, and $s_3 \geq 0$, with the initial conditions of $\mu'_{0,0,0} = 1$, $\mu'_{1,0,0} = \mu_1$, $\mu'_{0,1,0} = \mu_2$, and $\mu'_{0,0,1} = \mu_3$.

Finally, the expression of μ'_{s_1,s_2,s_3} based on the identity of [5] is given by

$$\mu'_{s_1,s_2,s_3} = \sum_{\nu_1=0}^{s_1}\sum_{\nu_2=0}^{s_2}\sum_{\nu_3=0}^{s_3}\sum_{r=0}^{\lfloor s/2\rfloor}(-1)^{\nu_1+\nu_2+\nu_3}\binom{s_1}{\nu_1}\binom{s_2}{\nu_2}\binom{s_3}{\nu_3}\frac{\left(\frac{\boldsymbol{h}_\nu^{\mathrm{T}}\Sigma\boldsymbol{h}_\nu}{2}\right)^r(\boldsymbol{h}_\nu^{\mathrm{T}}\boldsymbol{\mu})^{s-2r}}{r!(s-2r)!}, \tag{14}$$

where $s = s_1 + s_2 + s_3$ and $\boldsymbol{h}_\nu = [s_1/2 - \nu_1,\ s_2/2 - \nu_2,\ s_3/2 - \nu_3]^{\mathrm{T}}$.

In Table 3, we report the number of terms required by different methods for computing μ'_{s_1,s_2,s_3}, assuming $s_1 \geq s_2 \geq s_3$. When counting the number of terms for (11), we add the number of terms that are needed to create the univariate moments of z_1 and bivariate moments of (z_2, z_3). Similarly, when counting the number of terms for (12), we also add the number of terms that are needed to create the univariate moments of z_1, z_2, and z_3.[3]

Table 3: Number of terms for computing μ'_{s_1,s_2,s_3}, $d = \max[0, s_2 + s_3 - s_1]$

Method	Number of Terms
Song and Lee (2015) (9)	$\sum_{i=0}^{s_2}\sum_{j=0}^{\min[s_2,s_3-i]}\sum_{k=0}^{\min[s_2-i,s_3-j]}\left\lfloor\frac{s_1-i-j+2}{2}\right\rfloor\left\lfloor\frac{s_2-i-k+2}{2}\right\rfloor\left\lfloor\frac{s_3-j-k+2}{2}\right\rfloor$
New Algorithm (11)	$2(s-3) + 3s_2s_3 + (s_2+1)(s_3+1) - \frac{d(d+1)}{2}$
New Algorithm (12)	$2(s-3) + \frac{(s_3+1)(s_3+2)(3s_2-s_3+3)}{6} - \left\lceil\frac{d(d+2)(2d+5)}{24}\right\rceil$
Willink (2005) (13)	$2(s-3) + 3(s_1s_2 + s_1s_3 + s_2s_3) + 4s_1s_2s_3$
Kan (2008) (14)	$\lfloor(s_1+1)(s_2+1)(s_3+1)/2\rfloor\lfloor s/2+1\rfloor$

For comparison, Table 4 presents the number of terms for different methods under different combinations of (s_1, s_2, s_3). From Table 4, we can see that our two new algorithms, (11) and (12) generally requiring summing

[3] Details on the derivation of the number of terms in Table 3 are available upon request.

up fewer terms than other competing methods. This is particularly the case when s_1 is large. However, even for small (s_1, s_2, s_3), our new methods still offer substantial improvement over existing methods. For example, when $s_1 = s_2 = s_3 = 4$, (11) requires summing up 81 terms and (12) requires summing up 60 terms, whereas the other methods require summing up from 213 to 434 terms. Comparing (11) with (12), we see that the number of terms are comparable in most cases, but when (s_1, s_2, s_3) is large, (11) requires summing up far fewer terms than (12).

Table 4: Number of terms for computing μ'_{s_1,s_2,s_3} under different combinations of (s_1, s_2, s_3)

	Number of Terms				
(s_1, s_2, s_3)	(9)	(11)	(12)	(13)	(14)
(1,1,1)	4	6	4	13	8
(4,1,1)	12	13	11	49	40
(4,4,1)	45	33	25	148	125
(4,4,4)	213	81	60	418	434
(10,1,1)	27	25	23	121	154
(10,4,1)	117	46	38	346	440
(10,4,4)	624	103	85	958	1370
(10,10,1)	396	87	67	796	1331
(10,10,4)	2365	207	174	2182	3926
(10,10,10)	10836	420	435	4954	10640
(100,1,1)	252	205	203	1201	10504
(100,4,1)	1197	226	218	3316	26765
(100,4,4)	6834	283	265	9058	69410
(100,10,1)	4986	268	248	7546	62216
(100,10,4)	32062	397	367	20542	161066
(100,10,10)	179711	655	740	46534	372710
(100,100,1)	262701	897	697	70996	1030301
(100,100,4)	1883605	2097	1884	192802	2626706
(100,100,10)	13166076	4470	6735	436414	5947130
(100,100,100)	3321449001	35745	262020	4090594	77787650

Ultimately, what matters most is the speed of computing μ'_s for various algorithms. We do not report the CPU time for computing μ'_s in Table 4 because there are many cases that such computation is infeasible for the earlier algorithms as they involve a very large number of terms. To illustrate the benefit of (11) relative to the older algorithms, we consider a case with $\boldsymbol{s} = (s_1, 5, 5)$ for $5 \leq s_1 \leq 100$. In Fig. 1, we plot the ratio of CPU time of $\mu'_{s_1,5,5}$ of (13), (14) and (9) to the CPU time of (11) as a function of s_1.[4] As we can see from Fig. 1, the new algorithm (11) significantly outperforms existing algorithms in terms of computation time. In addition, the advantage increases with s_1. This is because in our new algorithm, we only need to compute $\mathrm{E}[z_1^i]$ for $0 \leq i \leq s_1$ and $\mathrm{E}[z_2^j z_3^k]$ for $0 \leq j \leq 5$ and $0 \leq k \leq 5$. Since the computation time of moments of z_1 only grows linearly with s_1 and the computation time of the bivariate moments of (z_2, z_3) is of $O(s_2 s_3)$, our new algorithm (11) significantly outperforms the other algorithms which typical has computation time that is of $O(s_1 s_2 s_3)$ or even higher.

3. New Algorithms for the General Case

Before we present our new algorithms for computing μ'_s for the general n case, we first introduce some notation. For a vector of nonnegative integers $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_n)$, we define $|\boldsymbol{\nu}| = \sum_{i=1}^n \nu_i$, $\boldsymbol{\nu}! = \prod_{i=1}^n \nu_i!$. For two vectors $\boldsymbol{\nu}$ and

[4]We implement all the algorithms in Matlab. The programs are run on a Ryzen 5950x PC. All the Matlab programs for the current paper are available at `https://www-2.rotman.utoronto.ca/~kan/research.htm`.

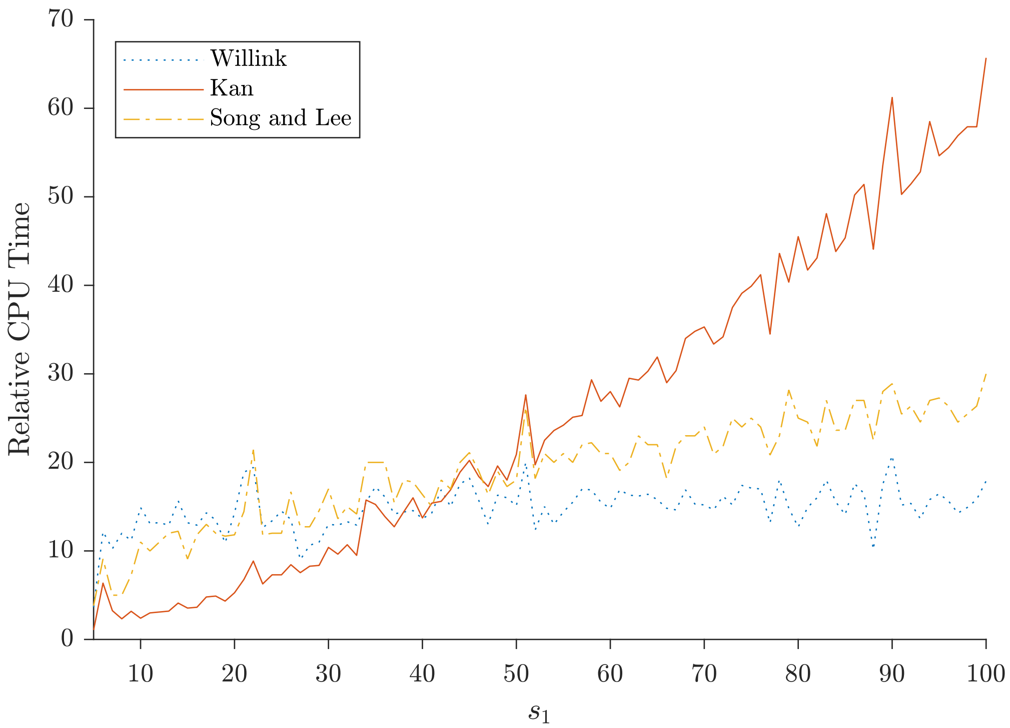

Fig. 1: Ratio of CPU time of computing $\boldsymbol{\mu}'_{s_1,5,5}$ using three different methods, relative to (11)

$\boldsymbol{s} = (s_1, \ldots, s_n)$, $\boldsymbol{\nu} \leq \boldsymbol{s}$ is a short-hand notation for $\nu_i \leq s_i$ for $i = 1, \ldots, n$, and

$$\binom{\boldsymbol{s}}{\boldsymbol{\nu}} = \prod_{i=1}^{n} \binom{s_i}{\nu_i}.$$

We now provide a quick summary of existing algorithms for computing $\mu'_{\boldsymbol{s}}$. For general n, [9] provides an $(n+1)$-term recursive relation for $\mu'_{s_1,\ldots,s_n}$:

$$\mu'_{s_1+1,\ldots,s_n} = \mu_1 \mu'_{s_1,\ldots,s_n} + \sum_{\substack{j=1 \\ s_j>0}}^{n} s_j \sigma_{1j} \mu'_{s_1,\ldots,s_j-1,\ldots,s_n}, \tag{15}$$

with the initial conditions of $\mu'_{\boldsymbol{0}} = 1$ and $\mu'_{\boldsymbol{e}_i} = \mu_i$, where $\boldsymbol{e}_i$ is an n-vector of zeroes, except that its i-th element is equal to one.

[5] provides the following explicit expression for $\mu'_{\boldsymbol{s}}$:

$$\mu'_{\boldsymbol{s}} = \sum_{\boldsymbol{0}\leq\boldsymbol{\nu}\leq\boldsymbol{s}} \sum_{r=0}^{\lfloor |s|/2 \rfloor} (-1)^{|\boldsymbol{\nu}|} \binom{\boldsymbol{s}}{\boldsymbol{\nu}} \frac{\left(\frac{\boldsymbol{h}_{\boldsymbol{\nu}}^{\mathrm{T}} \boldsymbol{\Sigma} \boldsymbol{h}_{\boldsymbol{\nu}}}{2}\right)^r (\boldsymbol{h}_{\boldsymbol{\nu}}^{\mathrm{T}} \boldsymbol{\mu})^{s-2r}}{r!(s-2r)!}, \tag{16}$$

where $\boldsymbol{h}_{\boldsymbol{\nu}} = [s_1/2 - \nu_1, \ldots, s_n/2 - \nu_n]^{\mathrm{T}}$.

[7] provides the following explicit formula of $\mu'_{\boldsymbol{s}}$

$$\mu'_{\boldsymbol{s}} = \sum_{\boldsymbol{l}\in\mathcal{L}_{\boldsymbol{s}}} d_{\boldsymbol{s},\boldsymbol{l}} \left(\prod_{i=1}^{n} \prod_{j=i}^{n} \sigma_{ij}^{l_{ij}} \right) \left(\prod_{i=1}^{n} \mu_i^{L_{\boldsymbol{s},i}} \right), \tag{17}$$

where $\boldsymbol{l} = (l_{ij})_{1\leq i\leq n, i\leq j\leq n}$ is a vector of $n(n+1)/2$ nonnegative integers,[5]

$$\begin{aligned} L_{\boldsymbol{s},i} &= s_i - \sum_{j=1}^{i-1} l_{ji} - \sum_{j=i+1}^{n} l_{ij} - 2l_{ii}, \\ d_{\boldsymbol{s},\boldsymbol{l}} &= \frac{\boldsymbol{s}!}{2^{\sum_{i=1}^{n} l_{ii}} \boldsymbol{l}! \prod_{i=1}^{n} L_{\boldsymbol{s},i}!}, \end{aligned} \tag{18}$$

and $\mathcal{L}_{\boldsymbol{s}}$ stands for the set of $\boldsymbol{l}$'s such that $L_{\boldsymbol{s},i} \geq 0$ for $i = 1, \ldots, n$.

We now provide a decomposition of $d_{\boldsymbol{s},\boldsymbol{l}}$ that allows us to express $\mu'_{\boldsymbol{s}}$ as a sum of various products of univariate moments of z_1 and multivariate moments of $(z_2, \ldots, z_n)$. Let

$$\begin{aligned} \boldsymbol{s}_2 &= (s_2, \ldots, s_n), \\ \boldsymbol{l}_1 &= (l_{12}, \ldots, l_{1n}), \\ \boldsymbol{h} &= (l_{ij})_{2\leq i\leq n, i\leq j\leq n}. \end{aligned}$$

By writing $\boldsymbol{l} = (l_{11}, \boldsymbol{l}_1, \boldsymbol{h})$, we can decompose $d_{\boldsymbol{s},\boldsymbol{l}}$ as

$$d_{\boldsymbol{s},\boldsymbol{l}} = \frac{\boldsymbol{s}!}{2^{l_{11}} l_{11}! (s_1 - |\boldsymbol{l}_1| - 2l_{11})! \boldsymbol{l}_1!} \frac{1}{2^{\sum_{i=2}^{n} l_{ii}} \boldsymbol{h}! \prod_{i=1}^{n-1} L_{\boldsymbol{s}_2-\boldsymbol{l}_1,i}!} = \frac{\boldsymbol{s}!}{(s_1 - |\boldsymbol{l}_1|)!(\boldsymbol{s}_2 - \boldsymbol{l}_1)! \boldsymbol{l}_1!} d_{s_1-|\boldsymbol{l}_1|, l_{11}} d_{\boldsymbol{s}_2-\boldsymbol{l}_1, \boldsymbol{h}},$$

where[6]

$$L_{\boldsymbol{s}_2-\boldsymbol{l}_1,i} = s_{i+1} - l_{1,i+1} - \sum_{j=2}^{i} l_{j,i+1} - \sum_{j=i+2}^{n} l_{i+1,j} - 2l_{i+1,i+1} = L_{\boldsymbol{s},i+1}, \quad i = 1, \ldots, n-1.$$

[5] $L_{\boldsymbol{s},i}$ is also a function of $\boldsymbol{l}$ but we suppress this dependence in its notation for convenience.

[6] It should be noted that $L_{\boldsymbol{s}_2-\boldsymbol{l}_1,i}$ depends on $\boldsymbol{h}$, whereas $L_{\boldsymbol{s},i}$ depends on $\boldsymbol{l}$.

Using this decomposition, we can write

$$
\begin{aligned}
\mu'_{\boldsymbol{s}} &= \sum_{\boldsymbol{l}\in\mathcal{L}_{\boldsymbol{s}}} \frac{\boldsymbol{s}!}{(s_1-|\boldsymbol{l}_1|)!(\boldsymbol{s}_2-\boldsymbol{l}_1)!\boldsymbol{l}_1!} d_{s_1-|\boldsymbol{l}_1|,l_{11}} d_{\boldsymbol{s}_2-\boldsymbol{l}_1,\boldsymbol{h}} \sigma_{11}^{l_{11}} \mu_1^{s_1-|\boldsymbol{l}_1|-2l_{11}} \left(\prod_{i=2}^n \sigma_{1i}^{l_{1i}}\right)\left(\prod_{i=2}^n\prod_{j=i}^n \sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{n-1} \mu_{i+1}^{L_{\boldsymbol{s}_2-\boldsymbol{l}_1,i}}\right) \\
&= \sum_{\substack{\boldsymbol{l}_1\leq \boldsymbol{s}_2\\ |\boldsymbol{l}_1|\leq s_1}} \frac{\boldsymbol{s}!}{(s_1-|\boldsymbol{l}_1|)!(\boldsymbol{s}_2-\boldsymbol{l}_1)!\boldsymbol{l}_1!}\left(\prod_{i=2}^n \sigma_{1i}^{l_{1i}}\right)^{\lfloor (s_1-|\boldsymbol{l}_1|)/2\rfloor} \sum_{l_{11}=0} d_{s_1-|\boldsymbol{l}_1|,l_{11}} \sigma_{11}^{l_{11}}\mu_1^{s_1-|\boldsymbol{l}_1|-2l_{11}} \\
&\quad \times \sum_{\boldsymbol{h}\in\mathcal{L}_{\boldsymbol{s}_2-\boldsymbol{l}_1}} d_{\boldsymbol{s}_2-\boldsymbol{l}_1,\boldsymbol{h}} \left(\prod_{i=2}^n\prod_{j=i}^n \sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{n-1} \mu_{i+1}^{L_{\boldsymbol{s}_2-\boldsymbol{l}_1,i}}\right) \\
&= \sum_{\substack{\boldsymbol{l}_1\leq \boldsymbol{s}_2\\ |\boldsymbol{l}_1|\leq s_1}} \frac{\boldsymbol{s}!}{(s_1-|\boldsymbol{l}_1|)!(\boldsymbol{s}_2-\boldsymbol{l}_1)!\boldsymbol{l}_1!}\left(\prod_{i=2}^n \sigma_{1i}^{l_{1i}}\right) \mathrm{E}\left[z_1^{s_1-|\boldsymbol{l}_1|}\right]\mathrm{E}\left[z_2^{s_2-l_{12}}\cdots z_n^{s_n-l_{1n}}\right].
\end{aligned}
$$

Writing $\boldsymbol{\nu} = (\nu_2,\ldots,\nu_n) = \boldsymbol{l}_1$ and $k = |\boldsymbol{l}_1|$, we obtain the following expression of $\mu'_{\boldsymbol{s}}$:

$$
\frac{\mu'_{\boldsymbol{s}}}{\boldsymbol{s}!} = \sum_{k=0}^{\min[s_1,|\boldsymbol{s}_2|]} \frac{\mathrm{E}\left[z_1^{s_1-k}\right]}{(s_1-k)!} \sum_{\substack{|\boldsymbol{\nu}|=k\\ \boldsymbol{\nu}\leq\boldsymbol{s}_2}} \frac{\prod_{i=2}^n \sigma_{1,i}^{\nu_i}}{\boldsymbol{\nu}!} \frac{\mathrm{E}\left[z_2^{s_2-\nu_2}\cdots z_n^{s_n-\nu_n}\right]}{(\boldsymbol{s}_2-\boldsymbol{\nu})!}.
$$

Let $\boldsymbol{\kappa} = \boldsymbol{s}_2 - \boldsymbol{\nu} = (\kappa_2,\ldots,\kappa_n)$, we can also write the above expression as

$$
\frac{\mu'_{\boldsymbol{s}}}{\boldsymbol{s}!} = \sum_{\substack{\boldsymbol{\kappa}:\boldsymbol{\kappa}\leq\boldsymbol{s}_2\\ \max(0,|\boldsymbol{s}_2|-s_1)\leq|\boldsymbol{\kappa}|}} \frac{\prod_{i=2}^n \sigma_{1,i}^{s_i-\kappa_i}}{(\boldsymbol{s}_2-\boldsymbol{\kappa})!} \frac{\mathrm{E}\left[z_1^{s_1-|\boldsymbol{s}_2-\boldsymbol{\kappa}|}\right]}{(s_1-|\boldsymbol{s}_2-\boldsymbol{\kappa}|)!} \frac{\mathrm{E}\left[z_2^{\kappa_2}\cdots z_n^{\kappa_n}\right]}{\boldsymbol{\kappa}!}. \tag{19}
$$

This expression allows us to express $\mu'_{\boldsymbol{s}}$ as a sum of various products of univariate moments of z_1 and multivariate moments of $(z_2,\ldots,z_n)$.

Repeating the above exercise, we can obtain an expression of $\mu'_{\boldsymbol{s}}$ that only depends on the univariate moments of z_i's. Let $\boldsymbol{b} = (b_{ij})_{1\leq i\leq n-1, i<j\leq n}$ be a vector of $n(n-1)/2$ nonnegative integers, we can write

$$
\frac{\mu'_{\boldsymbol{s}}}{\boldsymbol{s}!} = \sum_{\boldsymbol{b}\in\mathcal{B}_{\boldsymbol{s}}} \left(\prod_{i=1}^{n-1}\prod_{j=i+1}^n \frac{\sigma_{ij}^{b_{ij}}}{b_{ij}!}\right)\prod_{i=1}^n \frac{\mathrm{E}\left[z_i^{B_{\boldsymbol{s},i}}\right]}{B_{\boldsymbol{s},i}!}, \tag{20}
$$

where

$$
B_{\boldsymbol{s},i} = s_i - \sum_{j=1}^{i-1} b_{ji} - \sum_{j=i+1}^n b_{ij},
$$

and $\mathcal{B}_{\boldsymbol{s}}$ is the set of $\boldsymbol{b}$'s such that $B_{\boldsymbol{s},i}\geq 0$ for $i = 1,\ldots,n$. As an example, we can explicitly write $\sum_{\boldsymbol{b}\in\mathcal{B}_{\boldsymbol{s}}}$ for $n = 3$ as

$$
\sum_{b_{12}=0}^{\min[s_1,s_2]} \sum_{b_{13}=0}^{\min[s_3,s_1-b_{12}]} \sum_{b_{23}=0}^{\min[s_2-b_{12},s_3-b_{13}]}.
$$

Remark 1. When $s_1 = \ldots = s_n = 1$, (19) can be simplified to

$$
\mathrm{E}[z_1\ldots z_n] = \mathrm{E}[z_1]\mathrm{E}[z_2\ldots z_n] + \sigma_{12}\mathrm{E}[z_3\ldots z_n] + \sigma_{13}\mathrm{E}[z_2z_4\ldots z_n] + \ldots + \sigma_{1n}\mathrm{E}[z_1\ldots z_{n-1}],
$$

which is the same as Willink's recursive relation as given in (15).

Remark 2. When $s_1 = \ldots = s_n = 1$, (20) and (17) give the same expression. This is because under this case, we must have $l_{ii} = 0$ in order for $L_{\boldsymbol{s},i}\geq 0$. As a result, we have $L_{\boldsymbol{s},i} = B_{\boldsymbol{s},i}$ if we replace l_{ij} by b_{ij}.

Remark 3. When $\mu = \mathbf{0}$, $\mathrm{E}[z_2^{\kappa_2} \cdots z_n^{\kappa_n}]$ in (19) vanishes when $|\boldsymbol{\kappa}|$ is odd. Therefore, we can obtain $\mu'_{\boldsymbol{s}}$ with fewer terms. For even $|\boldsymbol{s}|$, we have

$$\frac{\mu'_{\boldsymbol{s}}}{\boldsymbol{s}!} = \sum_{k=\max[0,(|\boldsymbol{s}_2|-s_1)/2]}^{\lfloor |\boldsymbol{s}_2|/2 \rfloor} \sum_{\substack{\boldsymbol{\kappa}:\boldsymbol{\kappa}\leq \boldsymbol{s}_2 \\ |\boldsymbol{\kappa}|=2k}} \frac{\prod_{i=2}^n \sigma_{1,i}^{s_i-\kappa_i}}{(\boldsymbol{s}_2-\boldsymbol{\kappa})!} \frac{\mathrm{E}\left[z_1^{s_1-|\boldsymbol{s}_2-\boldsymbol{\kappa}|}\right]}{(s_1-|\boldsymbol{s}_2-\boldsymbol{\kappa}|)!} \frac{\mathrm{E}\left[z_2^{\kappa_2}\cdots z_n^{\kappa_n}\right]}{\boldsymbol{\kappa}!}.$$

Remark 4. Our decomposition formula (19) also offers a fast method for computing the number of terms in the explicit expression of $\mu'_{\boldsymbol{s}}$ in (17). Let $f(\boldsymbol{s})$ be the cardinality of $\mathcal{L}_{\boldsymbol{s}}$, we have[7]

$$f(\boldsymbol{s}) = \sum_{\substack{\mathbf{0}\leq\boldsymbol{\kappa}\leq \boldsymbol{s}_2 \\ \max[0,|\boldsymbol{s}_2|-s_1]\leq|\boldsymbol{\kappa}|}} \left[\left\lfloor \frac{s_1-|\boldsymbol{s}_2-\boldsymbol{\kappa}|}{2}\right\rfloor + 1\right] f(\boldsymbol{\kappa}).$$

Together with the boundary conditions of $f(\mathbf{0}) = 1$ and $f(s_1) = \lfloor s_1/2 \rfloor + 1$, the above recurrence relation allows us to obtain $f(\boldsymbol{s})$.

In general, (19) provides a much more efficient way of computing $\mu'_{\boldsymbol{s}}$ than the other methods. To illustrate this, we consider a number of examples in Table 5. From Table 5, we can see that (19) requires summing up the fewest number of terms, and the improvement is often substantial, especially when $|\boldsymbol{s}|$ is large.

Table 5: Number of terms for computing $\mu'_{\boldsymbol{s}}$ using different methods

Method	Number of terms			
	$\boldsymbol{s} = 5 \times 1_5$	$\boldsymbol{s} = (15,\ 1_{10})$	$\boldsymbol{s} = (40,\ 1_{10})$	$\boldsymbol{s} = 10 \times 1_5$
Willink (2005) (15)	40165	113641	292841	893090
Kan (2008) (16)	50544	106496	545792	20393650
Song and Lee (2015) (17)	684450	869483	2408818	807739076
New Algorithm (19)	5741	7175	7225	68891
New Algorithm (20)	62821	123127	123187	10757011

Counting the number of terms does not tell the complete story as the terms in each method would require different number of arithmetic operations. In Table 6, we report the CPU time of all the methods relative to that of (19) as well as the CPU time for (19). As can be seen from Table 6, there is a substantial improvement of computation speed for $\mu'_{\boldsymbol{s}}$ based on (19). In contrast, (17) and (20) perform the worst. Besides that these two methods require summing up a larger number of terms, the enumeration of $\mathcal{L}_{\boldsymbol{s}}$ and $\mathcal{B}_{\boldsymbol{s}}$ is also quite time consuming, which leads to a much lower execution speed for these two methods. Based on the results in Table 6 as well as the other experiments that we had performed, we recommend the use of (19) to compute $\mu'_{\boldsymbol{s}}$.[8]

In addition, we demonstrate the advantage of using (19) to compute $\mu'_{\boldsymbol{s}}$ over existing algorithms. We consider a case with $\boldsymbol{s} = (s_1, 5, 5, 5)$ for $5 \leq s_1 \leq 100$. In Fig. 2, we plot the ratio of CPU time of $\mu'_{s_1,5,5,5}$ of (15), (16) and (17) to the CPU time of (19) as a function of s_1. As we can see from Fig. 2, the new algorithm (19) significantly outperforms existing algorithms in terms of computation time. In addition, the advantage increases with s_1. Compared with the case of $n = 3$ in Fig. 1, the case of using (19) is even more compelling for $n = 4$.[9]

[7]This follows because for two different $\boldsymbol{\kappa}$'s, the explicit formula of $\mu'_{\boldsymbol{\kappa}}$ have no terms in common.

[8]All the algorithms involve summing up a number of terms and that can lead to cancellation error. This is particularly a concern for (16) because the terms involved are often of opposite signs. For the other algorithms, whether cancellation errors occur or not depend on the signs of μ_i and σ_{ij}. Nevertheless, we find that (19) and (20) produce almost identical answers to (15) and (17) in our experiment.

[9]Due to the scale of the graph, it is hard to read the ratio of CPU time for (15) to that of (19). This ratio goes up steadily from 1 to 16.8. In addition, we do not report the ratio of CPU time of using (20) in Fig. 2. It is in general slower to use (20) than (19), with an average ratio of CPU time of about 4.

Fig. 2: Ratio of CPU time of computing $\boldsymbol{\mu}'_{s_1,5,5,5}$ using three different methods, relative to (19)

Table 6: Ratio of CPU time for computing μ'_s using different methods, relative to (19)

Method	Relative CPU time			
	$s = 5 \times 1_5$	$s = (15,\ 1_{10})$	$s = (40,\ 1_{10})$	$s = 10 \times 1_5$
Willink (2005) (15)	1.5	7.5	18.8	5.2
Kan (2008) (16)	6.0	21.7	83.0	33.6
Song and Lee (2015) (17)	355.7	978.9	1431	103560
New Algorithm (20)	15.7	75.2	73.5	547.7
CPU Time of (19) (in seconds)	9.38e-5	8.13e-5	9.38e-5	7.53e-4

4. A General Decomposition Formula

In the last section, we provide two decomposition formulae for μ'_s. The first one is in terms of products of univariate moments of z_1 and multivariate moments of $(z_2, \dots, z_n)$. The second one is in terms of products of univariate moments of z_1 to z_n. It is natural to ask if there exists other ways of decomposing μ'_s. For example, can we express $\mathrm{E}[z_1^{s_1} z_2^{s_2} z_3^{s_3} z_4^{s_4}]$ in terms of product of bivariate moments of (z_1, z_2) and (z_3, z_4)? The following proposition answers the question in the affirmative. In particular, it provides a general formula that decomposes the product moments of z in terms of product of multivariate moments of two disjoint subsets of z.

Proposition 1. *Let m be a positive integer with $m < n$, we define*

$$\begin{aligned} s_a &= (s_1, \dots, s_m), \\ s_b &= (s_{m+1}, \dots, s_n), \\ l_i &= (l_{i,m+1}, \dots, l_{i,n}), \quad i = 1, \dots, m, \end{aligned}$$

where l_{ij}'s are nonnegative integers. Denote $\tilde{\mu}'_s = \mu'_s / s!$. We have

$$\tilde{\mu}'_s = \sum_{\substack{l_1 + \dots + l_m \leq s_b \\ |l_i| \leq s_i,\ i=1,\dots,m}} \left(\prod_{i=1}^{m} \prod_{j=m+1}^{n} \frac{\sigma_{ij}^{l_{ij}}}{l_{ij}!} \right) \tilde{\mu}'_{(s_c, 0_{n-m})} \tilde{\mu}'_{(0_m, s_d)}, \tag{21}$$

where $s_c = s_a - (|l_1|, \dots, |l_m|)$ and $s_d = s_b - l_1 - \dots - l_m$.

Proof. Let $h_1 = (l_{ij})_{1 \leq i \leq m, i \leq j \leq m}$ and $h_2 = (l_{ij})_{m+1 \leq i \leq n, i \leq j \leq n}$, we can write l in (18) as $l = (h_1, l_1, \dots, l_m, h_2)$. In addition, we define[10]

$$\begin{aligned} L_{s_c,i} &= s_i - |l_i| - \sum_{j=1}^{i-1} l_{j,i} - \sum_{j=i+1}^{m} l_{i,j} - 2 l_{i,i} = L_{s,i}, \quad i = 1, \dots, m, \\ L_{s_d,i} &= s_{m+i} - l_{1,m+i} - \dots l_{m,m+i} - \sum_{j=m+1}^{m+i-1} l_{j,m+i} - \sum_{j=m+i+1}^{n} l_{m+i,j} - 2 l_{m+i,m+i} = L_{s,m+i}, \quad i = 1, \dots, n-m. \end{aligned}$$

This allows us to decompose $d_{s,l}$ in (18) as

$$d_{s,l} = \frac{s!}{\prod_{i=1}^m l_i!} \times \frac{1}{2^{\sum_{i=1}^m l_{ii}} h_1! \prod_{i=1}^m L_{s_c,i}!} \times \frac{1}{2^{\sum_{i=m+1}^n l_{ii}} h_2! \prod_{i=1}^{n-m} L_{s_d,i}!} = \frac{s!}{s_c! s_d! \prod_{i=1}^m l_i!} d_{s_c,h_1} d_{s_d,h_2}.$$

[10]It should be noted that $L_{s_c,i}$ is based on h_1 and $L_{s_d,i}$ is based on h_2.

Using this decomposition, we obtain

$$
\begin{aligned}
\mu'_{\boldsymbol{s}} &= \sum_{\boldsymbol{l}\in\mathcal{L}_{\boldsymbol{s}}} d_{\boldsymbol{s},\boldsymbol{l}} \left(\prod_{i=1}^{n}\prod_{j=i}^{n}\sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{n}\mu_i^{L_{\boldsymbol{s},i}}\right)\\
&= \sum_{\boldsymbol{l}\in\mathcal{L}_{\boldsymbol{s}}} \frac{\boldsymbol{s}!}{\boldsymbol{s}_c!\boldsymbol{s}_d!\prod_{i=1}^{m}\boldsymbol{l}_i!} d_{\boldsymbol{s}_c,\boldsymbol{h}_1} d_{\boldsymbol{s}_d,\boldsymbol{h}_2} \left(\prod_{i=1}^{n}\prod_{j=i}^{n}\sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{m}\mu_i^{L_{\boldsymbol{s}_c,i}}\right)\left(\prod_{i=1}^{n-m}\mu_{m+i}^{L_{\boldsymbol{s}_d,i}}\right)\\
&= \sum_{\substack{\boldsymbol{l}_1+\cdots+\boldsymbol{l}_m\le \boldsymbol{s}_b\\ |\boldsymbol{l}_i|\le s_i,\ i=1,\dots,m}} \frac{\boldsymbol{s}!}{\boldsymbol{s}_c!\boldsymbol{s}_d!}\left(\prod_{i=1}^{m}\prod_{j=m+1}^{n}\frac{\sigma_{ij}^{l_{ij}}}{l_{ij}!}\right)\left[\sum_{\boldsymbol{h}_1\in\mathcal{L}_{\boldsymbol{s}_c}} d_{\boldsymbol{s}_c,\boldsymbol{h}_1}\left(\prod_{i=1}^{m}\prod_{j=i}^{m}\sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{m}\mu_i^{L_{\boldsymbol{s}_c,i}}\right)\right]\\
&\quad\times\left[\sum_{\boldsymbol{h}_2\in\mathcal{L}_{\boldsymbol{s}_d}} d_{\boldsymbol{s}_d,\boldsymbol{h}_2}\left(\prod_{i=m+1}^{n}\prod_{j=i}^{n}\sigma_{ij}^{l_{ij}}\right)\left(\prod_{i=1}^{n-m}\mu_{m+i}^{L_{\boldsymbol{s}_d,i}}\right)\right]\\
&= \boldsymbol{s}!\sum_{\substack{\boldsymbol{l}_1+\cdots+\boldsymbol{l}_m\le \boldsymbol{s}_b\\ |\boldsymbol{l}_i|\le s_i,\ i=1,\dots,m}}\left(\prod_{i=1}^{m}\prod_{j=m+1}^{n}\frac{\sigma_{ij}^{l_{ij}}}{l_{ij}!}\right)\frac{\mu'_{(\boldsymbol{s}_c,0_{n-m})}}{\boldsymbol{s}_c!}\frac{\mu'_{(0_m,\boldsymbol{s}_d)}}{\boldsymbol{s}_d!}.
\end{aligned}
$$

□

Proposition 1 suggests that we can express the product moments of $(z_1,\dots,z_n)$ in terms of product moments of $(z_1,\dots,z_m)$ and product moments of $(z_{m+1},\dots,z_n)$. When $m=1$, this nests the expression of (19) as a special case. Given that there are different ways of decomposing $\mu'_{\boldsymbol{s}}$, it would be ideal to figure out the optimal decomposition of $\mu'_{\boldsymbol{s}}$ that requires summing up the fewest number of terms. However, the optimal choice of m depends on the value of $\boldsymbol{s}$, and there is not a single decomposition that dominates the others for all values of $\boldsymbol{s}$. To illustrate this, we consider the case of $n=4$. For this case, we can use explicit formula (17) or the product of univariate moment formula (20). We can also use the decomposition formula (21) for $m=1$ to $m=3$ ($m=1$ is the same as (19)). In Table 7, we present the number of terms for these different methods under different values of (s_1,s_2,s_3,s_4). Compared with the explicit formula of (17), both (20) and (21) require summing up fewer number of terms. However, there is no clear-cut winner among these alternative algorithms. When $s_1=s_2=s_3=s_4$ (as in the first three cases), the decomposition formula of $\mu'_{\boldsymbol{s}}$ based on $m=2$ has the fewest number of terms. However, when (s_1,s_2,s_3,s_4) are not balanced, the other algorithms could involve summing up fewer number of terms. Since we consider cases with $s_1\ge s_2\ge s_3\ge s_4$ (which is without loss of generality as we can always rearrange the z_i's such that this condition is satisfied), we always find that when $s_1>s_4$, (21) with $m=1$ involves summing up fewer terms than (21) with $m=3$. This is easy to understand because univariate moments require fewer terms to compute than trivariate moments, so it is better to compute more univariate moments and less trivariate moments. When $m=1$, we only need to compute trivariate moments of (z_2,z_3,z_4) up to order of (s_2,s_3,s_4). However for $m=3$, we need to compute trivariate moments of (z_1,z_3,z_3) up to order of (s_1,s_2,s_3), and hence it is less efficient. For the general problem, further research is needed to identify the optimal algorithm for computing $\mu'_{\boldsymbol{s}}$ for a given value of $\boldsymbol{s}$.

5. Conclusion

The multivariate normal distribution is fundamental to mathematical statistics, and its moments play a central role in statistical methodology. For example, many statistical estimators can be written as quadratic forms in multivariate normal random variables, and their moments can be obtained by using the moments of product of multivariate normal random variables. In addition, the computation of product moments of multivariate normal random variables is closely related to the computation of multivariate Hermite polynomial (see [3] and [9]), and the latter have many useful applications in statistical theory, e.g., in Gram-Charlier expansions and Edgeworth approximations of distributions of sums of vector random variables.

There are also other applications that require the evaluation of $\mu'_{\boldsymbol{s}}$. For example, suppose z_i is the gross return (i.e., one plus net return) of an asset at time i. Then the terminal wealth of investing \$1 in the asset for n period is

$$W_n = z_1 z_2 \cdots z_n.$$

Table 7: Number of terms for computing μ'_{s_1,s_2,s_3,s_4} under different combinations of (s_1, s_2, s_3, s_4)

	Number of Terms				
			Decomposition Formula (21)		
(s_1, s_2, s_3, s_4)	(17)	(20)	$m = 1$	$m = 2$	$m = 3$
(2,2,2,2)	123	64	86	58	86
(4,4,4,4)	3810	665	459	275	459
(6,6,6,6)	49006	3656	1312	788	1312
(10,2,2,2)	667	109	119	127	326
(20,2,2,2)	1352	129	139	207	626
(20,20,2,2)	45138	962	823	1328	4730
(20,20,20,2)	3596802	23167	5387	2812	35726

For an investor with an investment horizon of n periods, he is interested in the moments of W_n, i.e.,

$$E[W_n^k] = E[z_1^k z_2^k \cdots z_n^k].$$

These moments are easy to compute when returns are independent over time. However, if returns are serially correlated, then the computation of $E[W_n^k]$ is far from trivial. If we are willing to assume $z \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the computation of $E[W_n^k]$ is the same as the computation of product moments of multivariate normal.

Therefore, an algorithm for the fast computation of μ'_s is of important value. While recursive and explicit formulae of μ'_s for the multivariate normality case are readily available in the statistics literature, they are often impractical for computational purpose when the number of random variables in the product is moderately large. In this paper, we present a new algorithm for computing μ'_s which provides a significant speed improvement over existing methods. Although the results in this paper are presented for the multivariate normality case, our algorithm can be extended to deal with the case that the random variables follow a multivariate elliptical distribution, along the line as suggested by [5].

Acknowledgments

We thank Joonsuk Huh and two anonymous referees for helpful comments.

References

[1] L. J. Bain, Moments of a noncentral t and noncentral F-distribution, American Statistician 23(4) (1969) 33–34.
[2] R. Blacher, Multivariate quadratic forms of random vectors, Journal of Multivariate Analysis 87 (2003) 2–23.
[3] B. Holmquist, The d-variate vector Hermite polynomial of order k. Linear Algebra and its Applications 237/238 (1996) 155–190.
[4] L. Isserlis, On a fomula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables, Biometrika 12 (1 and 2) (1918) 134–139.
[5] R. Kan, From moments of sum to moments of product, Journal of Multivariate Analysis 99 (2008) 542–554.
[6] T. H. Savits, Some statistical applications of Faa di Bruno, Journal of Multivariate Analysis 97 (2006) 2131–2140.
[7] I. Song, S. Lee, Explicit formulae for product moments of multivariate Gaussian random variables, Statistics and Probability Letters 100 (2015) 27–34.
[8] A. Takemura, K. Takeuchi, Some results on univariate and multivariate Cornish-Fisher expansion: algebraic properties and validity, Sankhyā: The Indian Journal of Statistics, Series A 50(1) (1988) 111-136.
[9] R. Willink, Normal moments and Hermite polynomials, Statistics and Probability Letters 73 (2005) 271–275.