Spring 1-1-2017

On Minimum Variance Unbiased Estimation of a Power of an Unknown Scalar or Matrix

Kathleen Elise Smith
University of Colorado at Boulder, kes256@gmail.com

On minimum variance unbiased estimation of a power of an unknown scalar or matrix

by

Kathleen Elise Smith

B.S., Norwich University, 2006

M.A., University of Colorado Boulder, 2014

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics

2017

This thesis entitled:
On minimum variance unbiased estimation of a power of an unknown scalar or matrix
written by Kathleen Elise Smith
has been approved for the Department of Mathematics

Prof. Sergei Kuznetsov

Prof. János Engländer

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Smith, Kathleen Elise (Ph.D., Mathematics)

On minimum variance unbiased estimation of a power of an unknown scalar or matrix

Thesis directed by Prof. Sergei Kuznetsov

This thesis extends work on finding optimal estimates of P^t, both in the case where P is a scalar, and when P is a matrix. In the scalar case, we present a formula for minimum variance unbiased estimates of P^t, given k independent observations of the scalar P. We discuss the generalization to the matrix case, and compare the new estimate to the unbiased estimate presented by Kuznetsov and Orlov (13). We compare the estimates in terms of variance and computation. This comparison is done both theoretically and computationally.

Acknowledgements

This research was supported by the SMART Scholarship.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Background and Motivation

We wish to find optimal estimates for P^t given k independent observations of P when P is a $d \times d$ matrix. In this case, we have interesting applictations. Suppose P represents a transition matrix for a discrete time Markov process. Then estimating P^t gives an estimate of the state of the process after t steps.

Markov chains have many applications, with applications in finance and reliability having particular relevance to the estimation of P^t. In finance and economics, a Markov model can be used to simulate and predict demand for products (6, chap. 6, 7), and more generally resource use and allocation. Credit ratings and the likelihood that a given bond transitions from one rating category to another are modelled as a Markov chain using historical data, and the predictions of these models are used to assess the risk of and price financial securities (6, chap. 7). Markov chains have been used to model and predict the overall state of the economy, most simply by classifying the state as either one of economic growth or decline, with various inferences from this state used in further models, as in Hamilton (9) and Calvet and Fisher (4). In Evans and Mueller (8), a Markov chain is used to model real estate cycles, with an estimated transition matrix being used to predict the likelihood of different points in the cycle occuring at given future times, explicitly using an estimate of P^t. Markov chains can be used to predict the effect of advertising campaigns on customers, as in Pfeifer and Carraway (20) and Ching and Ng (6, chap. 5), and have been used to model payouts of slot machines in Oses (18).

Markov chains can be used to model and predict genetic expression in cells (6, chap. 7), and have been used to denoise images in Catak (5). Markov chains are used in Natural Language Processing, with transition matrices estimated from different corpora of text, and to model the growth and composition of various populations. They have been used to model and predict infrastructure deterioration (12), and to model reliability of equipment in general. This has more relevance as server farms are more commonly used and maintained by various companies. It is surely of interest to these companies to be able to predict whether a critical number of servers may have failed in a given time period. Finally, while the above applications generally deal with situations in which a prediction about a particular future time is valuable, many more applications, such as Google's PageRank (19) deal with estimations of a transition matrix in which the steady state probabilities are most valuable.

Given these examples of using Markov chains to model a wide range of applications, there is also plenty of work done in estimating transition matrices in many of these applications. For instance, Nichols et al. (16) proposes and anayzes two methods to estimate a transition matrix in stage-based population projection models. Baik et al. (1) and Ortiz-García et al. (17) propose methods of estimating transition matrices to model wastewater systems and pavement deterioration. In Hu et al. (10), methods of estimating Markov transition matrices used to analyze risk associated an investment portfolio are extended from more conventional investments to a class of low credit quality issuers for which less historical information is available. Finally, many methods for estimating Markov tranistion matrices given incomplete observations of the process are proposed, such as that in Barsotti et al. (2). This does not even touch on Hidden Markov Models (HMMs), an entire field of work studying Markov models with hidden states. However, the question of how best to estimate P^t, as opposed to P, has not been answered.

When considering an estimate for P^t when P is a matrix, we cannot simply look for a minimum variance unbiased estimate (MVUE), as variance of a matrix is not a well defined or easy to work with concept. We can consider the individual elements of the matrix as a vector, and compute this vector's covariance matrix. Then we consider an estimate to be optimal in terms of

its variance if the covariance matrix is less than the covariance matrix associated with any other estimate. (We say the matrix A is less than B here, if $B - A$ is positive definite.) Unfortunately, this computation quickly becomes unmanageable for relatively small values of k and t (it was only attempted for $d = 2$). An alternate method of comparing estimates for a matrix is to require that the estimate is unbiased, and then compare the variance of a scalar version of the estimate. It is reasonable, though not certain, that the variance of the scalar estimate corresponds to the variance of the matrix entries in some positive way. For this reason, finding an MVUE in the scalar case is an interesting problem, and this thesis solves this previously open problem.

1.2 Previous Work

We wish to give minimum variance unbiased estimates (MVUEs) for P^t given k independent observations of P. First, we will assume P is a Gaussian scalar, with unknown mean and variance. Next, we will investigate how to extend an estimate to the case when P is an $d \times d$ matrix. In both cases, we will compare our work with previous results, which hold for both scalars and matrices.

Previously, Kuznetsov and Orlov (13) derived formulae for unbiased estimates, which holds when P is either a Gaussian scalar or a square matrix with Gaussian entries. This result makes clever use of complex numbers to come up with a compact, elegant formula, with lots of versatility. The following theorem gives a wide variety of unbiased estimates, though it can be generalized further to yield even more.

Theorem 1.1. *Let $\hat{P}_1, ..., \hat{P}_k$ be independent observations of P. Then if $\sum_{i=1}^{k} c_i = 0$ and $\sum_{i=1}^{k} c_i^2 = \frac{1}{k}$ for $c_i \in \mathbb{R}$,*

$$\left(\frac{\hat{P}_1 + \cdots + \hat{P}_k}{k} + i(c_1\hat{P}_1 + \cdots + c_k\hat{P}_k) \right)^t, \tag{1.1}$$

is unbiased for P^t.

Note that in practice, we will take the real part of the estimate, though its expectation is purely real anyway. Next, a minimum variance estimate is always symmetric in the observations, so we will always consider the symmetrization of the estimate. Finally, this factored form is much

more amenable than listing all possible terms of degree t and determining each term's coefficient. To list all possible degree t terms, we would need to list partitions of t (into at most k parts), which is non-trivial.

1.3 Initial Challenges

This brings us to the first challenges in finding our MVUE. First, sticking with the formula for an unbiased estimate, computing its variance is messy and does not simplify in general, given that we are dealing with a complex quantity. This is not conducive to either computing or proving very much about the variance. Second, if we avoid the issue of complex numbers, we are left with writing out the degree t terms of a polynomial, whose exponents correspond to partitions of t. Again, this is hard to generalize, as very little can be said about even the number of partitions, let alone writing them down.

Chapter 2

Scalar Case

2.1 Initial Attempts

The first attempts cover the case where P is a scalar. When treating P as a scalar, we will assume P is normally distributed with mean μ and variance σ^2.

The first approach was to use the unbiased estimate (1.1), and find a way for estimates of that form to achieve minimum variance. This work was done almost exclusively for $k = 3$. A few (unsuccessful) attempts to investigate $k = 4$ were made, in the hopes that patterns would show up there despite being unapparent for $k = 3$. For $k = 2$, there is only one, unique, unbiased estimate, which is automatically minimum variance.

By exhaustion, a form of (1.1) was found that gave the minimum variance estimate through $t = 12$, but no such form exists that works for $t = 13$. See Mathematica code in Appendix B.1.

2.2 Lehmann-Scheffé and Rao-Blackwell

Given that (1.1) did not yield MVUE for $t \geq 13$, a new approach was needed. Working with the expanded polynomial is intractable, but applying the Rao-Blackwell (21, 3) and Lehmann-Scheffé (15, 14) theorems allowed us to study a particular form of polynomial, and avoid computing variance altogether. These theorems require the following definitions, and the proofs will follow those presented in Cox and Hinkley (7), Roussas (22).

Definition 2.1 (Sufficient Statistic)**.** *Consider data X with probability distribution parametrized by*

θ. A statistic $T(X)$ is sufficient for θ if the conditional distribution of X given T does not depend on θ.

The Fisher-Neyman factorization theorem gives necessary and sufficient conditions for determining whether a statistic is sufficient.

Theorem 2.1 (Fisher-Neyman Factorization Theorem)**.** *Suppose the data $X = \{X_1, \ldots, X_k\}$ has probability distribution function $f_X(x_1, \ldots, x_k, \theta)$. Then a statistic $T(X)$ is sufficient for the parameter θ if and only if f_X factors into two components, $h(X)$ and $g_\theta(T, \theta)$, where h does not depend on θ, and g does not depend on X except as a function of T.*

See Appendix A.1 for proof.

Definition 2.2 (Complete Statistic)**.** *Consider data X with probability distribution parametrized by θ. A statistic $S(X)$ is complete for θ if for any function g such that $E[g(S(X))] = 0$ for all θ, $g(S(X)) = 0$ almost everywhere for all θ.*

Theorem 2.2 (Rao-Blackwell Theorem)**.** *Given an unbiased estimate for $g(\theta)$, $\varphi(X)$, and a sufficient statistic $T(X)$ of some parameters θ of the data X, the Rao-Blackwell estimator is $\hat{\varphi}(X) = E[\varphi(X)|T(X)]$. The Rao-Blackwell estimator is again unbiased for $g(\theta)$, and has variance no greater than the original estimator.*

See Appendix A.2 for proof.

Theorem 2.3 (Lehmann-Scheffé Theorem)**.** *Given the same hypotheses as in Rao-Blackwell, if we also assume $T(X)$ is a complete statistic for θ, $\hat{\varphi}(X)$ is the unique MVUE for $g(\theta)$.*

See Appendix A.3 for proof.

We wish to use these theorems to find an MVUE. Let $X_1, \ldots, X_k$ be iid $\sim \mathcal{N}(\mu, \sigma^2)$. Define $S_1 = \sum_{i=1}^k X_i/k$ and $\tilde{S}_2 = \sum_{i=1}^k X_i^2/k$. First, we will use Theorem 2.1 to show that $S_1, \tilde{S}_2$ is sufficient for μ and σ.

Proof. $X_1, \ldots, X_k$ have pdf

$$f(X_1, \ldots, X_k) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(X_i-\mu)^2}{2\sigma^2}}.$$

Since f is in fact a function of S_1 and $\tilde{S}_2$, it factors into $f = h(X_1, \ldots, X_k)g(S_1, \tilde{S}_2, \mu, \sigma)$ with $h = 1$.

$$\begin{aligned} f(X_1, \ldots, X_k) &= \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(X_i-\mu)^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}^k} e^{\frac{-\sum_{i=1}^{k}(X_i-\mu)^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}^k} e^{\frac{-\sum_{i=1}^{k}(X_i^2-2X_i\mu+\mu^2)}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}^k} e^{\frac{-\sum X_i^2+2\sum X_i\mu-k\mu^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}^k} e^{\frac{-k\tilde{S}_2+2kS_1\mu-k\mu^2}{2\sigma^2}}. \end{aligned}$$

□

We leave proof of completeness for the next section, in which we adjust $\tilde{S}_2$. The initial attempts to find an unbiased estimate of μ^t of the form

$$\varphi_t(X) = \sum_{i=0}^{\lfloor t/2 \rfloor} \tilde{c}_i S_1^{t-2i} \tilde{S}_2^i$$

resulted in the following conjecture:

Conjecture 2.4. *Setting* $\tilde{c}_i = \binom{t}{2i} \frac{(2i-1)!!(-1)^i(k-3)!!}{(k+2i-3)!!}$,

$$\varphi_t(X) = \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{2i} \frac{(2i-1)!!(-1)^i((k+2t-2i-4)!!(k-3)!!}{(k+t-3)!} S_1^{t-2i} \tilde{S}_2^i \tag{2.1}$$

is an unbiased (and therefore minimum variance) estimate.

This was found by computing the expectation of the formula with fixed t and general k, from which the part of the formula depending on k can be deduced, and then comparing this for different values of t to deduce the rest of the formula. For the values of t for which this conjecture has been confirmed, it holds for all k. See Appendix B.2 for Mathematica code. Attempts to

proove this conjecture for all t were unsuccessful. Some combinatoric interpretation of the estimate was possible, but not helpful. The main difficulty in finding a general proof lies in taking the expectation of $S_1^{t-2i}\tilde{S}_2^i$, which involves first expanding all terms. In the next section, we adjust $\tilde{S}_2$ so that S_1 and S_2 are independent.

2.3 Adjustment to $\tilde{S}_2$

In the next approach, we adjust $\tilde{S}_2$. Let $S_2 = \sum_{i=1}^k (X_i - S_1)^2/k = \tilde{S}_2 - S_1^2$. (S_1, S_2) is an invertible function of $(S_1, \tilde{S}_2)$, so it follows from the sufficiency of $(S_1, \tilde{S}_2)$ that (S_1, S_2) is sufficient as well. Let $\varphi_t(X) = \sum_{i=0}^{\lfloor t/2 \rfloor} c_i S_1^{t-2i} S_2^i$. We will show that S_1 and S_2 and independent, determine their distributions, and prove that (S_1, S_2) is a complete statistic.

Lemma 2.5. *S_1 and S_2 are independent.*

Proof. Considering that S_1 and $\{X_i - S_1\}$ form a multivariate normal vector, we will compute $\mathrm{Cov}(S_1, X_i - S_1) = E[(S_1 - \mu)(X_i - S_1)]$. Then

$$\begin{aligned} E[(S_1 - \mu)(X_i - S_1)] &= E[S_1 X_i - S_1^2 - \mu(X_i - S_1)] \\ &= \frac{1}{k} E[X_i(X_1 + \cdot + X_k)] - \frac{1}{k^2} E[(X_1 + \cdots + X_k)(X_1 + \cdots + X_k)] \\ &= \frac{1}{k}(E[X_1^2] + (k-1)\mu^2) - \frac{1}{k^2}(kE[X_1^2] + k(k-1)\mu^2) \\ &= \frac{\mu^2 + \sigma^2}{k} + \frac{k-1}{k}\mu^2 - \frac{\mu^2 + \sigma^2}{k} + \frac{k-1}{k}\mu^2 \\ &= 0. \end{aligned}$$

Since S_1 and $(X_i - S_1)$ are independent for each i, S_1 and $S_2 = g((X_1 - S_1), \ldots, (X_k - S_1))$ are independent as well. □

Next, we can find the expectations of S_1^{t-2i} and S_2^i. S_1 is a Gaussian, with mean μ and variance σ^2/k, so $E[S_1^j] = \sum_{j=0}^{\lfloor (t-2i)/2 \rfloor} \binom{t-2i}{2j} (2j-1)!! \mu^{t-2i-2j} \frac{\sigma^{2j}}{k^j}$.

Lemma 2.6. *$\frac{kS_2}{\sigma^2}$ is a χ^2 variable with $k-1$ degrees of freedom. As such,*

$E[S_2^i] = \frac{\sigma^{2i}}{k^i} \prod_{j=1}^{i} (k + 2j - 3)$.

Proof. Consider the iid standard normal random variables $Z_1, \ldots, Z_k$. Then $\sum_{i=1}^k (Z_i - \bar{Z})^2 + k\bar{Z}^2 = Z_1^2 + \cdots + Z_k^2 \sim \chi^2(k)$, where $\bar{Z}$ is the mean of the Z_i. The covariance calculation in Lemma 2.5 still holds for the Z_i, so $\sum_{i=1}^k (Z_i - \bar{Z})^2$ and $k\bar{Z}^2$ are also independent. Further, $k\bar{Z}^2 = (\sqrt{k}\bar{Z})^2$ has a $\chi^2(1)$ distribution. Finally, we can use the independence of $\sum_{i=1}^k (Z_i - \bar{Z})^2$ and $k\bar{Z}^2$ to find the moment generating function of $\sum_{i=1}^k (Z_i - \bar{Z})^2$, which we will denote $M_Z(t)$. The moment generating function of a $\chi^2(k)$ is given by $(1-2t)^{-k/2}$, so we have $(1-2t)^{-k/2} = M_Z(t)(1-2t)^{-1/2}$, and $M_Z(t) = (1-2t)^{-(k-1)/2}$. So we see that $\sum_{i=1}^k (Z_i - \bar{Z})^2 \sim \chi^2(k-1)$. From here, we see that

$$\begin{aligned} S_2 &= \sum_{i=1}^k (\sigma(Z_i + \mu) - \sigma(\bar{Z} + \mu))^2/k \\ &= \sum_{i=1}^k (\sigma(Z_i - \bar{Z}))^2/k \\ &= \frac{\sigma^2}{k} \sum_{i=1}^k (Z_i - \bar{Z})^2. \end{aligned}$$

Thus $\frac{kS_2}{\sigma^2}$ is a $\chi^2(k-1)$. □

Lemma 2.7. *Let $X_1, \ldots, X_k$ be iid $\sim \mathcal{N}(\mu, \sigma^2)$. Define $S_1 = \sum_{i=1}^k X_i/k$ and $S_2 = \sum_{i=1}^k (X_i - S_1)^2/k$. Then (S_1, S_2) is complete for μ and σ.*

Proof. Let $f(S_1, S_2)$ be any function for which $E[f(S_1, S_2)] = 0$. Separating f into f^+ and f^-, we can write

$$\begin{aligned} 0 &= E[f] \\ &= \int_{-\infty}^{\infty} \int_0^{\infty} (f^+(S_1, S_2) - f^-(S_1, S_2)) \frac{e^{\frac{-(S_1-\mu)^2}{2\sigma^2/k}}}{\sqrt{2\pi(\sigma^2/k)}} \frac{e^{\frac{-kS_2}{2\sigma^2}}}{2^{(k-1)/2}\Gamma\left(\frac{k-1}{2}\right)} \left(\frac{kS_2}{\sigma^2}\right)^{(k-1)/2} \frac{k}{\sigma^2}\, dS_2\, dS_1 \\ &= \int_{-\infty}^{\infty} \int_0^{\infty} (f^+(S_1, S_2) - f^-(S_1, S_2)) \left(\frac{kS_2}{\sigma^2}\right)^{(k-1)/2} e^{\frac{-S_1^2}{2\sigma^2/k}} e^{\frac{S_1\mu}{\sigma^2/k}} e^{\frac{-kS_2}{2\sigma^2}}\, dS_2\, dS_1. \end{aligned}$$

Let $g(S_1, S_2) = \left(\frac{kS_2}{\sigma^2}\right)^{(k-1)/2} e^{\frac{-S_1^2}{2\sigma^2/k}}$. Then

$$\int_{-\infty}^{\infty} \int_0^{\infty} f^+(S_1, S_2) g(S_1, S_2) e^{\frac{S_1\mu}{\sigma^2/k}} e^{\frac{-kS_2}{2\sigma^2}}\, dS_2\, dS_1 = \int_{-\infty}^{\infty} \int_0^{\infty} f^-(S_1, S_2) g(S_1, S_2) e^{\frac{S_1\mu}{\sigma^2/k}} e^{\frac{-kS_2}{2\sigma^2}}\, dS_2\, dS_1.$$

But this is a 2-dimensional Laplace transform, so we must have

$$f^+(S_1, S_2) g(S_1, S_2) = f^-(S_1, S_2) g(S_1, S_2)$$

almost everywhere, from which it follows that $f = 0$ almost everywhere as required. □

Corollary 2.7.1. *Given k iid observations from a normal distribution with mean μ and variance σ^2 (both unknown), $X_1, \ldots, X_k$, let $S_1 = \sum_{i=1}^k X_i/k$ and $S_2 = \sum_{i=1}^k (X_i - S_1)^2/k$. Then any function of S_1 and S_2 which is unbiased for a function of μ, $f(\mu)$, is also the MVUE for $f(\mu)$.*

So we can look for an MVUE of the form

$$\varphi_t(X) = \sum_{i=0}^{\lfloor t/2 \rfloor} c_i S_1^{t-2i} S_2^i.$$

Theorem 2.8. *Setting $c_i = \binom{t}{2i} \frac{(2i-1)!!(-1)^i(k-3)!!}{(k+2i-3)!!}$,*

$$\varphi_t(X) = \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{2i} \frac{(2i-1)!!(-1)^i(k-3)!!}{(k+2i-3)!!} S_1^{t-2i} S_2^i \tag{2.2}$$

is an unbiased (and therefore minimum variance) estimate for μ^t.

Proof. We know $E[\varphi_t(X)]$ will have the form $\sum_{\ell=0}^{\lfloor t/2 \rfloor} a'_\ell \mu^{t-2\ell}$. In particular, we will see that it has the form $\sum_{\ell=0}^{\lfloor t/2 \rfloor} a_\ell \mu^{t-2\ell} \frac{\sigma^{2\ell}}{k^\ell}$. We need to show that $a_\ell = 1$ when $\ell = 0$, and is 0 otherwise. We will begin by computing $E[c_i S_1^{t-2i} S_2^i]$. S_1 and S_2 are independent, so the expectation will be straightforward. First, S_1 is normal with mean μ and variance $\frac{\sigma^2}{k}$. So, $E[S_1^{t-2i}] = \sum_{j=0}^{\lfloor (t-2i)/2 \rfloor} \binom{t-2i}{2j} (2j-1)!! \mu^{t-2i-2j} \frac{\sigma^{2j}}{k^j}$.

Next, as calculated above, $E[S_2^i] = \frac{\sigma^{2i}}{k^i} \prod_{j=1}^i (k + 2j - 3)$.

Combining these,

$$E[c_i S_1^{t-2i} S_2^i] = c_i \sum_{j=0}^{\lfloor (t-2i)/2 \rfloor} \binom{t-2i}{2j} (2j-1)!! \mu^{t-2i-2j} \frac{\sigma^{2i+2j}}{k^{i+j}} \prod_{b=1}^{i} (k + 2b - 3).$$

Whenever $i + j = \ell$, we get a term containing $\mu^{t-2\ell} \frac{\sigma^{2\ell}}{k^\ell}$. So, we wish to sum over terms where $i + j = \ell$.

$$a_\ell = \sum_{i+j=\ell} c_i \binom{t-2i}{2j} (2j-1)!! \prod_{b=1}^{i} (k + 2b - 3).$$

We need to show $a_0 = 1$, and $a_\ell = 0$ otherwise. $a_0 = c_0$ here, and according to the conjectured formula, $c_0 = 1$. For $a_\ell, \ell > 0$, we will substitute in the conjectured formula for c_i.

$$\begin{aligned}
a_\ell &= \sum_{i+j=\ell} c_i \binom{t-2i}{2j} (2j-1)!! \prod_{b=1}^{i} (k+2b-3) \\
&= \sum_{i=0}^{\ell} c_i \binom{t-2i}{2(\ell-i)} (2(\ell-i)-1)!! \prod_{b=1}^{i} (k+2b-3) \\
&= \sum_{i=0}^{\ell} (-1)^i \binom{t}{2i} \frac{(2i-1)!!}{\prod_{b=1}^{i}(k+2b-3)} \binom{t-2i}{2(\ell-i)} (2(\ell-i)-1)!! \prod_{b=1}^{i} (k+2b-3) \\
&= \sum_{i=0}^{\ell} (-1)^i \binom{t}{2i} (2i-1)!! \binom{t-2i}{2(\ell-i)} (2(\ell-i)-1)!!
\end{aligned}$$

For each i, we are counting the ways to first choose i pairs from t elements, then choose an additional $\ell - i$ pairs from the remaining $t - 2i$ elements, for a total of ℓ pairs. Let's say the first i pairs are blue, and the remainder are red, as the two sets of pairs are distinct. The number of ways to do this is the same as if we first chose ℓ pairs from the t elements, then out of the ℓ pairs, chose i of them to be blue, and the remainder to be red. Rewriting a_ℓ, we then get

$$\begin{aligned}
a_\ell &= \sum_{i=0}^{\ell} (-1)^i \binom{t}{2\ell} (2\ell-1)!! \binom{\ell}{i} \\
&= \binom{t}{2\ell} (2\ell-1)!! \sum_{i=0}^{\ell} (-1)^i \binom{\ell}{i} \\
&= 0,
\end{aligned}$$

since $\sum_{i=0}^{\ell} (-1)^i \binom{\ell}{i} = 0$ for any ℓ, and the estimate is unbiased. □

2.4 Cramér-Rao bound on variance

A conjectured formula for the variance of this unbiased estimate is

$$\sum_{i=1}^{n} \binom{n}{i}^2 \frac{i!(k+2i-4)!!(k-3)!!}{k^i(k+i-3)!} \mu^{2n-2i} \sigma^{2i}.$$

This formula for the variance has been verified for all k, for t up to $t = 13$. See Appendix B.3 for Mathematica code.

The Cramér-Rao bound on the variance is $\frac{t^2}{k}\mu^{2t-2}\sigma^2$. See Appendix A.4 for derivation. Mathematica code in Appendix B.4 show that for t up to 15, the estimate (2.2) is asymptotically efficient. However, the estimate is not fully efficient for any t for which variance was calculated.

Finally, although we have not proven any particular formula for the variance for general t, through analysis of the variance we can show that the estimate is asymptotically efficient for all t.

Theorem 2.9. *The estimate* (2.2) *is asymptotically efficient for all values of* t *and* k *when* P *is a Gaussian random variable.*

Proof. Consider $E[(\varphi_t(X) - P^t)^2] = E[\varphi_t(X)^2] - P^{2t}$.

$$\begin{aligned} E[\varphi_t(X)^2] &= \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{j=0}^{\lfloor t/2 \rfloor} c_i c_j E[S_1^{2t-2i-2j} S_2^{i+j}] \\ &= \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{j=0}^{\lfloor t/2 \rfloor} c_i c_j \prod_{a=1}^{i+j} (k+2a-3) \sum_{m=0}^{t-i-j} \binom{2t-2i-2j}{2m} (2m-1)!! \frac{\mu^{2t-2i-2j-2m}\sigma^{i+j+m}}{k^{i+j+m}}. \end{aligned}$$

Let $\ell = i + j + m$. Then

$$E[\varphi_t(X)^2] = \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{j=0}^{\lfloor t/2 \rfloor} c_i c_j \prod_{a=1}^{i+j} (k+2a-3) \sum_{m=0}^{t-\ell+m} \binom{2t-2\ell+2m}{2m} (2m-1)!! \frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}}. \tag{2.3}$$

Note that for a given ℓ, the coefficient of $\frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}}$ is given by

$$b_\ell = \sum_{i+j+m=\ell} c_i c_j \prod_{a=1}^{i+j} (k+2a-3) \binom{2t-2\ell+2m}{2m} (2m-1)!!.$$

Next, note that for $\ell = 0$, the coefficient of $\frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}} = \mu^{2t}$ in $E[\varphi_t(X)^2]$ is given when $i = j = m = 0$, and is equal to 1. We are interested in $E[\varphi_t(X)^{2t}] - \mu^{2t}$, which we can slightly simplify:

$$\begin{aligned} E[\varphi_t(X)^{2t}] - \mu^{2t} &= \sum_{\ell=0}^{t} b_\ell \frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}} - \mu^{2t} \\ &= \mu^{2t} + \sum_{\ell=1}^{t} b_\ell \frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}} - \mu^{2t} \\ &= \sum_{\ell=1}^{t} b_\ell \frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}}. \end{aligned}$$

Now in order to show the estimate is asymptotically efficient, we must show

$$\lim_{k\to\infty} \frac{\sum_{\ell=1}^{t} b_\ell \frac{\mu^{2t-2\ell}\sigma^{\ell}}{k^{\ell}}}{\frac{t^2}{k}\mu^{2t-2}\sigma^2} = 1.$$

To do this, first consider b_ℓ. We will show that b_ℓ is asymptotically constant in k, so that all terms except for $\ell = 1$ vanish in the limit. To see that b_ℓ is asymptotically constant in k, note that $c_i = \binom{t}{2i}\frac{(2i-1)!!(-1)^i(k-3)!!}{(k+2i-3)!!}$ has i more factors of k in the denominator than the numerator. So $c_i c_j$ will contribute $i+j$ factors of k to the denominator in the limit. This exactly counters the product $\prod_{a=1}^{i+j}(k+2a-3)$, so that b_ℓ is asymptotically constant in k.

Finally, for $\ell = 1$, we must show that $b_1 = t^2$.

$$\begin{aligned} b_1 &= \sum_{i+j+m=1} c_i c_j \prod_{a=1}^{i+j}(k+2a-3)\binom{2t-2+2m}{2m}(2m-1)!! \\ &= 2\left(c_0 c_1 (k-1)\right) + c_0^2\binom{2t}{2}. \end{aligned}$$

The second line consists of the two (identical) terms where either i or j is 1, and the other indices are 0, and of the term where $m = 1$ and $i = j = 0$.

Substituting in $c_0 = 1$ and $c_1 = -\frac{t^2-t}{2(k-1)}$, we get

$$\begin{aligned} 2\left(c_0 c_1 (k-1)\right) + c_0^2\binom{2t}{2} &= 2\left(-\frac{t^2-t}{2(k-1)}(k-1)\right) + \binom{2t}{2} \\ &= \binom{2t}{2} - (t^2-t) \\ &= \frac{2t(2t-1)}{2} - \frac{2(t^2-t)}{2} \\ &= \frac{4t^2-2t-2t^2+2t}{2} \\ &= \frac{2t^2}{2} \\ &= t^2. \end{aligned}$$

$b_1 = t^2$, so the estimate is asymptotically efficient. □

Chapter 3

Matrix Case

3.1 Extension to Matrices

First, we must introduce some notation.

Definition 3.1 (Order Averaging)**.** *Given square matrices $A_1, \ldots, A_k$, let $A_1^{s_1} \otimes \cdots \otimes A_k^{s_k}$ be the average of all possible orderings of s_1 copies of A_1, through s_k copies of A_k. Let t be the sum $s_1 + \cdots + s_k$. Note that if $A_1, \ldots, A_k$ commute,*

$$A_1^{s_1} \otimes \cdots \otimes A_k^{s_k} = A_1^{s_1} \cdots A_k^{s_k}.$$

Note that the average is the same whether we consider multiple copies of each A_i distinct or not.

Lemma 3.1. *Given square matrices $P, D_1, \ldots, D_k$ and integers $t_1, \ldots, t_k$ that sum to t,*

$$(P + D_1)^{t_1} \otimes \cdots \otimes (P + D_k)^{t_k} \tag{3.1}$$

is equal to

$$\sum_{\ell=0}^{t} \sum_{\sum s_i = \ell} \binom{t_1}{s_1} \cdots \binom{t_k}{s_k} P^{t-\ell} \otimes D_1^{s_1} \otimes \cdots \otimes D_k^{s_k} \tag{3.2}$$

Proof. Let us first assign labels to the factors in (1): A_1 through A_{t_1} are each equal to $(P + D_1)$, A_{t_1+1} through $A_{t_1+t_2}$ are each equal to $(P + D_2)$, all the way through A_{t-t_k} to A_t which are equal to $(P + D_k)$. We will consider the (noncommutative) product $A_1 \cdots A_t$. First, we will consider a single term from the expansion, $A_1^{\gamma_1} \cdots A_t^{\gamma_t}$, where γ_i indicates whether we selected P or D_j from

the factor A_i. Now, consider making the same selections γ_i in every ordering of the product of the A_i. This term will show up in every possible order (and be averaged) in the expansion of $A_1 \otimes \cdots \otimes A_t$. This holds for every selection of γ_i, so we know $A_1 \otimes \cdots \otimes A_t = \sum A_1^{\gamma_1} \otimes \cdots \otimes A_t^{\gamma_t}$, where we sum over all possibe selections of the γ_i. However, since $\otimes$ is a commutative operation, we can simplify this to $(P + D_1)^{t_1} \otimes \cdots \otimes (P + D_k)^{t_k} = \sum_{l=0}^{t} \sum_{\sum s_i = l} \binom{t_1}{s_1} \cdots \binom{t_k}{s_k} P^{t-l} \otimes D_1^{s_1} \otimes \cdots \otimes D_k^{s_k}$, using the binomial theorem. □

We will also make use of the following Lemma and proof, originally by Kuznetsoz and Orlov:

Lemma 3.2 (Kuznetsov and Orlov (13))**.** *Given non-negative integers $s, s_1, \ldots, s_k$ and iid square $d \times d$ matrices with multivariate Gaussian distribution and expectation zero, $A_1, \ldots, A_t, t = s_1 + \cdots + s_k$, and a deterministic matrix P of the same dimension,*

$$EP^s \otimes A_1^{2s_1} \otimes \cdots \otimes A_r^{2s_k} = (2s_1 - 1)!! \ldots (2s_k - 1)!! EP^s \otimes A_1^2 \otimes \cdots \otimes A_t^2 \tag{3.3}$$

We begin with some notation. For a finite set M, denote by $|M|$ the number of elements in M. Let $N = \{m_1, \ldots, m_n\}, m_1 < \cdots < m_n$ be a set of natural numbers and let $s_1, \ldots, s_k \geq 0$ be some integers such that $s_1 + s_2 + \cdots + s_k \leq n$. Denote by $\Lambda(s_1, \ldots, s_k, N)$ a collection of all partitions of the set N into disjoint subsets $M_1, \ldots, M_k$ and the remainder $M_0 = N \setminus \cap_{i=1}^k M_i$, such that $|M_1| = s_1, \ldots, |M_k| = s_k$. For an element $\mu = (M_1, \ldots, M_k) \in \Lambda(s_1, \ldots, s_k, N)$ and $l \in N$ we set

$$\mu(l) = \sum_{j=1}^{k} j I_{M_j}(l) \tag{3.4}$$

where I_A is the indicator function, so that $\mu(l) = j$ if $l \in M_j, j = 0, 1, \ldots, k$. The function $\mu(l)$ determines the partition μ uniquely. In a special case $s_1 = \cdots = s_k = 2, 2k \leq n$ we denote

$$\Lambda^*(k, N) = \Lambda(2, 2, \ldots, 2, N)$$

Finally, if $N = \{1, 2, \ldots, n\}$, we set

$$\Lambda_n(s_1, \ldots, s_k) = \Lambda(s_1, \ldots, s_k, N), \quad \Lambda_n^*(k) = \Lambda^*(k, N)$$

The role of this notation is as follows. Let $s, s_1, \dots, s_k$ be non-negative integers and let $n = s + 2s_1 + \dots + 2s_k$. Then

$$A_0^s \otimes A_1^{s_1} \otimes \dots \otimes A_k^{s_k} = \frac{s!(2s_1)! \dots (2s_k)!}{n!} \sum_{\mu \in \Lambda_n(2s_1,\dots,2s_k)} A_{\mu(1)} A_{\mu(2)} \dots A_{\mu(n)} \tag{3.5}$$

For a partition $\mu = \{M_1, \dots, M_k\} \in \Lambda^*(k, N)$, yet another function $\mu^*(l), l = 1, 2, \dots, n$ can be defined. Namely, the values $\mu^*(1), \dots, \mu^*(2k)$ are defined by the condition

$$M_i = \{\mu^*(2i-1), \mu^*(2i)\}, \mu^*(2i-1) < \mu^*(2i), i = 1, \dots, k$$

and the rest of values comes from the condition

$$M_0 = \{\mu^*(2k+1), \dots, \mu^*(n)\}, \mu^*(2k+1) < \mu^*(2k+2) < \dots < \mu^*(n)$$

Clearly, the function μ^* is defined uniquely and, if μ^* and k is known, the partition μ can be found.

The proof of Lemma 3.2 is based on the following technical statement.

Lemma 3.3. *Let $s_1, \dots, s_k$ be non-negative integers such that $2(s_1 + \dots + s_k) \le n$ and let*

$$\delta_i = (\delta_i^1, \dots, \delta_i^n), i = 1, 2, \dots, s_1 + \dots + s_k$$

be a sequence of independent identically distributed vectors that have a multivariate Gaussian distribution with zero expectation. Also, let

$$\delta_0 = (\delta_0^1, \dots, \delta_0^n)$$

be a non-random vector. Then

$$\begin{aligned} E \sum_{\mu \in \Lambda_n(2s_1,\dots,2s_k)} & \delta_{\mu(1)}^1 \delta_{\mu(2)}^2 \dots \delta_{\mu(n)}^n \\ & = \frac{1}{s_1! \dots s_k!} E \sum_{\mu \in \Lambda_n^*(s_1+\dots+s_k)} \delta_{\mu(1)}^1 \delta_{\mu(2)}^2 \dots \delta_{\mu(n)}^n \end{aligned} \tag{3.6}$$

Note that, in the left part of the formula (3.6) only $k+1$ vectors $\delta_0, \delta_1, \dots, \delta_k$ are used and on the right we have all the vectors $\delta_0, \delta_1, \dots, \delta_{s_1+\dots+s_k}$

Proof of Lemma 3.3. We will use Wick's theorem (also known as Isserlis' Theorem, see for example Isserlis (11)). Let $X_1, \dots, X_{2m}$ be a sequence of Gaussian random variables with zero expectation. Then the expectation of the product of all of them could be found in terms of their pairwise covariances. Namely,

$$EX_1 \dots X_{2m} = \frac{1}{m!} \sum_{\mu \in \Lambda^*_{2m}(m)} \prod_{i=1}^{m} \mathrm{Cov}(X_{\mu^*(2i-1)}, X_{\mu^*(2i)}). \tag{3.7}$$

With (3.7) in mind, fix some $\mu = \{M_1, \dots, M_k\} \in \Lambda_n(2s_1, \dots, 2s_k)$. Since the vectors δ_i are independent, we have

$$E\delta^1_{\mu(1)}\delta^2_{\mu(2)} \dots \delta^n_{\mu(n)} = \prod_{j \in M_0} \delta^j_0 \; E \prod_{j \in M_1} \delta^j_1 \; \dots \; E \prod_{j \in M_k} \delta^j_k$$

Now, let $M_1 = (d_1, \dots, d_{2s_1})$. By (3.7)

$$E \prod_{j \in M_1} \delta^j_1 = \frac{1}{s_1!} \sum_{\nu_1 \in \Lambda^*(s_1, M_1)} \prod_{i=1}^{s_1} \mathrm{Cov}(\delta_1^{\nu_1^*(d_{2i-1})}, \delta_1^{\nu_1^*(d_{2i})}) = \frac{1}{s_1!} \sum_{\nu_1 \in \Lambda^*(s_1, M_1)} E \prod_{j \in M_1} \delta^j_{\nu_1(j)}$$

(note that $\mathrm{Cov}(\delta^1_i, \delta^1_j) = E\delta^1_i\delta^1_j$ since the vector δ^1 has zero expectation). In the same way, we get

$$E \prod_{j \in M_2} \delta^j_2 = \frac{1}{s_2!} \sum_{\nu_2 \in \Lambda^*(s_2, M_2)} E \prod_{j \in M_2} \delta^j_{\nu_2(j)+s_1}$$

(note that the vectors $\delta^{s_1+1}, \dots, \delta^{s_1+s_2}$ are used on the right), and so on. Finally,

$$\begin{aligned}
&E \sum_{\mu \in \Lambda_n(2s_1, \dots, 2s_r)} \delta^1_{\mu(1)}\delta^2_{\mu(2)} \dots \delta^n_{\mu(n)} \\
&= \frac{1}{s_1! \dots s_k!} \sum_{\mu \in \Lambda_n(2s_1, \dots, 2s_k)} \sum_{\nu_1 \in \Lambda^*(s_1, M_1)} \cdots \sum_{\nu_r \in \Lambda^*(s_k, M_k)} \\
&\qquad E \prod_{j \in M_0} \delta^j_0 \prod_{j \in M_1} \delta^j_{\nu_1(j)} \prod_{j \in M_2} \delta^j_{\nu_2(j)+s_1} \cdots \prod_{j \in M_k} \delta^j_{\nu_k(j)+s_1+s_2+\dots+s_{k-1}} \\
&= \frac{1}{s_1! \dots s_k!} \sum_{\mu \in \Lambda^*_n(s_1+\dots+s_k)} E \prod_{j=1}^{n} \delta^j_{\mu(j)}
\end{aligned}$$

The last identity holds, because, choosing first a partition $\mu = \{M_1, \dots, M_k\}$ of the set $N = \{1, 2, \dots, n\}$ into the subsets $M_1, \dots, M_k$ and the remainder M_0, and then, for each i, choosing a partition ν_i of the set M_i into pairs $\hat{M}^i_1, \dots, \hat{M}^i_{s_i}$, we establish a one-to-one correspondence between the set of all possible combinations of partitions $\mu, \nu_1, \dots, \nu_k$, and the set $\Lambda^*_n(s_1 + \dots + s_k)$ of

partitions of N into $s_1 + \cdots + s_k$ pairs and a remainder M_0. Namely, to a combination $\mu, \nu_1, \ldots, \nu_k$ there corresponds a partition

$$\lambda = \{\hat{M}_1^1, \ldots \hat{M}_{s_1}^1, \hat{M}_1^2, \ldots, \hat{M}_1^k, \ldots, \hat{M}_{s_k}^k\}.$$

□

Proof of Lemma 3.2. Let $n = s + 2s_1 + \cdots + 2s_k$. Denote by $(B)^{ij}$ an element of the matrix B indexed by i and j. Also, denote $A_0 = P$. By Lemma 3.3 and by (3.5),

$$\begin{aligned}
&E(A_0^s \otimes A_1^{2s_1} \otimes \cdots \otimes A_r^{2s_k})^{i_0 i_n} \\
&\quad = \frac{s!(2s_1)! \ldots (2s_k)!}{n!} E \sum_{\mu \in \Lambda_n(2s_1, \ldots, 2s_k)} (A_{\mu(1)} \ldots A_{\mu(n)})^{i_0 i_n} \\
&\quad = \frac{s!(2s_1)! \ldots (2s_k)!}{n!} \sum_{i_1=1}^{d} \cdots \sum_{i_{n-1}=1}^{d} E \sum_{\mu \in \Lambda_n(2s_1, \ldots, 2s_k)} A_{\mu(1)}^{i_0 i_1} A_{\mu(2)}^{i_1 i_2} \ldots A_{\mu(n)}^{i_{n-1} i_n} \\
&\quad = \frac{s!(2s_1)! \ldots (2s_k)!}{n! s_1! \ldots s_k!} \sum_{i_1=1}^{d} \cdots \sum_{i_{n-1}=1}^{d} E \sum_{\lambda \in \Lambda_n^*(s_1 + \cdots + s_k)} A_{\lambda(1)}^{i_0 i_1} A_{\lambda(2)}^{i_1 i_2} \ldots A_{\lambda(n)}^{i_{n-1} i_n} \\
&\quad = \frac{s!(2s_1)! \ldots (2s_k)!}{n! s_1! \ldots s_k!} \frac{n!}{s! 2^{s_1 + \cdots + s_k}} E(A_0^s \otimes A_1^2 \otimes \cdots \otimes A_{s_1 + \cdots + s_k}^2)^{i_0 i_n} \\
&\quad = \frac{(2s_1)! \ldots (2s_k)!}{s_1! \ldots s_k! 2^{s_1 + \cdots + s_k}} E(A_0^s \otimes A_1^2 \otimes \cdots \otimes A_{s_1 + \cdots + s_k}^2)^{i_0 i_n} \\
&\quad = (2s_1 - 1)!! \ldots (2s_k - 1)!! E(A_0^s \otimes A_1^2 \otimes \cdots \otimes A_{s_1 + \cdots + s_k}^2)^{i_0 i_n}
\end{aligned}$$

□

Note that this Lemma holds for commutative matrices, and in the case where the matrices are dimension 1, ie scalars, reduces exactly to Wick's or Isserlis' Theorem.

In order to extend the scalar MVUE, we will introduce order averaging to the observations of a matrix P, $\hat{P}_1, \ldots, \hat{P}_k$, to account for the non-commutative matrix products. We wish to apply Lemma 3.2, so we will also adjust the $\hat{P}_i$ in order to deal with mean zero objects. We will let $\hat{P}_i = P + D_i$, but will not apply this until after applying the order averaging. In order to apply

order averaging to the observations, we will first expand S_1 and S_2 in the scalar estimate, $\varphi_t(X)$.

$$\begin{aligned}\varphi_t(X) &= \sum_{i=0}^{\lfloor t/2 \rfloor} c_i S_1^{t-2i} S_2^i \\ &= \sum_{i=0}^{\lfloor t/2 \rfloor} c_i S_1^{t-2i} (\tilde{S}_2 - S_1^2)^i \\ &= \sum_{i=0}^{\lfloor t/2 \rfloor} c_i S_1^{t-2i} \sum_{m=0}^{i} \binom{i}{m} \tilde{S}_2^m S_1^{2i-2m} (-1)^{i-m} \\ &= \sum_{i=0}^{\lfloor t/2 \rfloor} c_i \sum_{m=0}^{i} (-1)^{i-m} \binom{i}{m} S_1^{t-2m} \tilde{S}_2^m.\end{aligned}$$

Next, we substitute in $S_1 = \sum_{i=0}^{k} \frac{X_i}{k}$, $\tilde{S}_2 = \sum_{i=0}^{k} \frac{X_i^2}{k}$.

$$\varphi_t(X) = \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} (X_1 + \cdots + X_k)^{t-2m} (X_1^2 + \cdots + X_k^2)^m.$$

Now, for fixed m, we can expand

$$(X_1 + \cdots + X_k)^{t-2m} = \sum_{\sum q_i = t-2m} \tbinom{t-2m}{q_1, \ldots, q_k} X_1^{q_1} \cdots X_k^{q_k}$$

and

$$(X_1^2 + \cdots + X_k^2)^m = \sum_{\sum r_i = m} \tbinom{m}{r_1, \ldots, r_k} X_1^{2r_1} \cdots X_k^{2r_k}.$$

Putting this together we have

$$\begin{aligned}&(X_1 + \cdots + X_k)^{t-2m} (X_1^2 + \cdots + X_k^2)^m \\ &\qquad = \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} X_1^{q_1+2r_1} \cdots X_k^{q_k+2r_k}.\end{aligned}$$

Now we can apply order averaging to these terms to create a matrix version of the estimate, $\varphi_t(\hat{P})$.

Theorem 3.4. *Given k independent observations of P, $\hat{P}_1, \ldots, \hat{P}_k$,*

$$\varphi_t(\hat{P}) = \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} \hat{P}_1^{q_1+2r_1} \otimes \cdots \otimes \hat{P}_k^{q_k+2r_k} \tag{3.8}$$

is an unbiased estimate for P^t, in the case where P is a $d \times d$ square matrix with multivariate Gaussian entries.

Proof. We wish to apply Lemma 3.2, so we first rewrite $\hat{P}_i = D_i + P$.

$$E[\varphi_t(\hat{P})] = \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} \cdot E[\hat{P}_1^{q_1+2r_1} \otimes \cdots \otimes \hat{P}_k^{q_k+2r_k}]$$

$$= \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} \cdot E[(D_1 + P)^{q_1+2r_1} \otimes \cdots \otimes (D_k + P)^{q_k+2r_k}]$$

Now, by Lemma 3.1, we can expand the product $(D_1 + P)^{q_1+2r_1} \otimes \cdots \otimes (D_k + P)^{q_k+2r_k}$. We will also eliminate any term containing $D_i^{s_i}$ where s_i is odd, because the odd moments of the D_i are all 0.

$$= \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} \cdot \sum_{l=0}^{t} \sum_{\sum s_i = l} \binom{q_1+2r_1}{s_1} \cdots \binom{q_k+2r_k}{s_k} E[P^{t-l} \otimes D_1^{s_1} \otimes \cdots \otimes D_k^{s_k}]$$

$$= \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} \cdot \sum_{l=0}^{\lfloor t/2 \rfloor} \sum_{\sum s_i = l} \binom{q_1+2r_1}{2s_1} \cdots \binom{q_k+2r_k}{2s_k} E[P^{t-2l} \otimes D_1^{2s_1} \otimes \cdots \otimes D_k^{2s_k}]$$

and we can finally apply Lemma 3.2

$$= \sum_{i=0}^{\lfloor t/2 \rfloor} \sum_{m=0}^{i} \sum_{\sum q_i = t-2m} \sum_{\sum r_i = m} (-1)^{i-m} \binom{i}{m} \frac{c_i}{k^{t-m}} \binom{t-2m}{q_1, \ldots, q_k} \binom{m}{r_1, \ldots, r_k} \cdot \sum_{l=0}^{\lfloor t/2 \rfloor} \sum_{\sum s_i = l} \binom{q_1+2r_1}{2s_1} \cdots \binom{q_k+2r_k}{2s_k} (2s_1 - 1)!! \cdots (2s_k - 1)!! E[P^{t-2l} \otimes D_1^2 \otimes \cdots \otimes D_l^2].$$

Now, we wish to show that the coefficient of $E[P^{t-2l} \otimes D_1^2 \otimes \cdots \otimes D_l^2]$ is 1 when $l = 0$ and 0 otherwise. Recalling that everything we have done holds for scalars as well as matrices, note that

for scalars,

$$E[P^{t-2l} \otimes D_1^2 \otimes \cdots \otimes D_l^2] = E[P^{(t-2l)} D_1^2 \cdots D_l^2]$$
$$= P^{t-2l}\sigma^{2l}.$$

Since the original formula is unbiased for scalars, it is clear that we must have the coefficient of $E[P^{t-2l} \otimes D_1^2 \otimes \cdots \otimes D_l^2]$, which is the coefficient of $P^{t-2l}\sigma^{2l}$ in the case of scalars, equal to 1 when $l = 0$ and 0 otherwise. So we see that this estimate is also unbiased for matrices. □

Finally, one note on the formula is that since

$$S_1^{t-2i} \otimes S_2^i \neq \sum_{\sum q_j + 2r_j = t-2i} b_{q,r,i} \hat{P}_1^{q_1+2r_1} \otimes \cdots \otimes \hat{P}_k^{q_k+2r_k},$$

where each $\hat{P}_i$ is taken q_i times from S_1 and r_i times from S_2, and $b_{q,r,i}$ is the appropriate multinomial coefficient for each term, we cannot write the formula for matrices in terms of S_1 and S_2. In fact,

$$\hat{P} = \sum_{i=0}^{\lfloor t/2 \rfloor} c_i S_1^{t-2i} \otimes S_2^i$$

is a biased estimate. This is computationally inconvenient as well as notationally messy.

Chapter 4

Conclusions

4.1 Comparison of Estimates

We wish to determine the optimal way to estimate P^t when P is a matrix. It is computationally difficult to compare estimates in this case, so we will first compare estimates for scalars. Consider an estimate of the form

$$\mathrm{Re}\left[\left(\bar{X} + \mathrm{i}(\sum_{j=1}^{k} c_j X_j)\right)^t\right], \tag{4.1}$$

where $\bar{X}$ is the mean of the X_i. We can expand it and take the real part:

$$\begin{aligned}\mathrm{Re}\left[\left(\bar{X} + \mathrm{i}(\sum_{j=1}^{k} c_j X_j)\right)^t\right] &= \mathrm{Re}\left[\sum_{i=0}^{t} \binom{t}{i} \bar{X}^{t-i} \mathrm{i}^i (\sum_{j=1}^{k} c_j X_j)^i\right] \\ &= \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{2i} \bar{X}^{t-2i} (-1)^i (\sum_{j=1}^{k} c_j X_j)^{2i}.\end{aligned}$$

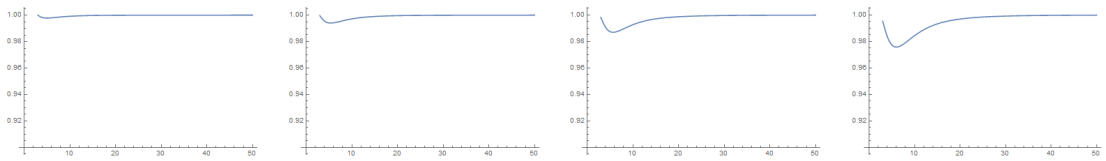
It is interesting to note that this now has a similar (though not identical) form to the MVUE (2.2). If we substitute $c\sqrt{\sum_{i=1}^{k} X_i^2/k}$ for the linear combination of the X_i, we would in fact get $\sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{2i} (-1)^i c^{2i} \bar{X}^{t-2i} S_2^i$. In order for this to be unbiased, c^{2i} would need to equal $\frac{(k-3)!!}{(k+2i-3)!!}$, which is not possible in a general formula. This is indicative that an estimate of the form (4.1) will not generally match the minimum variance estimate, though it is possible to find estimates of this form that coincide with the MVUE for small t. For $k = 3$, $t > 12$, it was possible to find coefficients in (4.1) such that this estimate coincided with the MVUE for scalars. For $t = 12$, this is not possible. This was only tested explicitly for $k = 3$, but it seems likely that for larger k, minimum variance estimates of the form (4.1) are also possible for small t.

4.1.1 Asymptotic Efficiency

In Kuznetsov and Orlov (13), the symmetrization of

$$\left(\bar{X} + i\left(\frac{X_i - X_j}{\sqrt{2k}}\right)\right)^t \tag{4.2}$$

is recommended for estimating P^t with k observations. This is easily generalized to any number of observations, and in computing its variance, has the advantage that $\bar{X}$ and $X_i - X_j$ are independent. Based on our work, this is not generally the minimum variance estimate for scalars, but it is asymptotically efficient for t up to at least $t = 6$. This was found with Mathematica code in Appendix B.5. Figure 4.1 plots the efficiency of (4.2) relative to (2.2). In plots of the efficiencies of (4.2) and (2.2) relative to the Cramér-Rao bound for $t < 10$, no difference is visible in the default Mathematica plot. For this reason, the plots in Figure 4.1 show $\mathrm{Var}(2.2)/\mathrm{Var}(4.2)$. For $t < 6$, the estimates coincide, and for $t > 9$, the computation of their variances was quite slow. In the plots of efficiency, the y-axis ranges from 0.9 to 1 and the number of observations k ranges from 3 to 50. As noted in Kuznetsov and Orlov (13), the efficiency of (4.2) depends on the ratio σ/μ, and this ratio was 1 ($\mu = \sigma = 1$) in calculating the plotted efficiency. Given a larger ratio, one would not expect to be able to make very reliable estimates, and given a smaller ratio, the difference between the estimates is even less. From Figure 4.1, we see that (4.2) is very close in efficiency to (2.2), even for small values of k, though the difference between the two estimates grows as t increases.



(a) $t = 6$ (b) $t = 7$ (c) $t = 8$ (d) $t = 9$

Figure 4.1: Efficiency of (4.2) relative to (2.2) for several t

4.2 Comparison of Computation

As mentioned in Section 3.1, the estimate given by (3.8) is computationally inconvenient. For relatively small k and t, it quickly becomes prohibitively complex. For example, for $t = 12$ and $k = 3$, the order-averaged polynomial consists of over 500,000 terms. Compare this to (4.2), which for any t, when $k = 3$, consists of 3 terms. To further compare the practicality of the two estimates, the Euclid computer (a Dell R730 with 128GB of memory and 2x Intel Xeon E5-2630 v3 2.4GHz processors) was used to compute each estimate for randomly generated observations. Table 4.1 shows the average time it took to compute the two estimates on Euclid, for 2×2 matrices, when $k = 3$. For each t, three random 2×2 matrices were generated, with real entries between 0 and 1, and then used to construct the two different estimates. This was repeated 10 times, and the average computation time was recorded. The maximum and minimum times were also recorded, but there were no outliers, so only the averages are reported here. See Appendix B.7 for Mathematica code. From this, it is clear that (4.2) is much more practical to compute for large t.

Table 4.1: Time to compute estimates

t	Time to compute (3.8) (s)	Time to compute (4.2) (s)
3	0.001714	0.000723
4	0.003519	0.000698
5	0.006901	0.000702
6	0.024232	0.000660
7	0.086110	0.000806
8	0.219614	0.000794
9	0.866936	0.000846
10	2.731224	0.000801
11	7.614402	0.000852
12	29.668838	0.000890
13	96.202817	0.000876
14	254.281603	0.000844

4.3 Comparison of Matrix Estimates

We conjecture that the estimate in Theorem 3.4 is optimal in the sense that the difference between the covariance matrix of the matrix entries in any unbiased estimate and the covariance matrix of the matrix entries in (3.8) is positive semi-definite. The main theoretical result in this area that we calculated is that for a 2×2 matrix, with $k = 3$ and $t = 6$, the difference between the covariance matrix of the matrix entries in (4.2) and the covariance matrix of the matrix entries in (3.8) is indeed positive definite. See Mathematica code in Appendix B.8.

In this computation, t and k were chosen as the lowest values such that the two estimates differ in the scalar case. For $k = 3$, $t < 6$, the estimates are identical, and the covariance matrices associated with them are (as expected) also the same. For $t = 6$, the variance of the scalar estimates differ by $\frac{5\sigma^{12}}{162}$. Generalized to 2×2 matrices, the eigenvalues of the difference between the covariances matrices associated with the estimates are $\frac{10\sigma^{12}}{81}$ and $\frac{25\sigma^{12}}{486}$ (with multiplicity 3). A similar computation was started for $t = 7$, but Mathematica has been unable to determine whether the difference between the covariance matrices is positive definite. These were computed with the assumption that the entries in the matrix were independant and distinct, but the variance for each entry was the same, σ^2.

This result is indicative that MVUE for scalars corresponds to an optimal estimate for matrices with respect to variance. However, considering the additional computation required for (3.8), the improvement over any non-optimal estimate given by Theorem 1.1 is not necessarily worth it.

4.4 Recommendations

In comparing the two estimates, (3.8) and (4.2), we have found that they seem very close in terms of variance for the scalar case, and that (4.2) is more computationally practical in the matrix case. We expect that the difference in variance in the scalar case extends to the matrix case in some way. Based on these observations, we recommend (4.2) as a practical estimate of P^t.

It is still unknown whether (3.8) is the optimal estimate in terms of variance for matrices, or

even whether (3.8) compares to (4.2) in the same way for matrices as it does for scalars. While S_1 and S_2 are not complete sufficient statistics when extended to matrices, it would be interesting to determine whether any statisticcs could play that role in the matrix case.

It could also be worth investigating the efficiency of thee two estimates. If it is possible to find a way to compute (3.8) more efficiently, it may be a better overall estimate than (4.2). This seems unlikely, and perhaps some measure of computational efficiency could quantify the difference and show that (4.2) is truly better in this sense.

Bibliography

[1] Hyeon-Shik Baik, Hyung Seok Jeong, and Dulcy M Abraham. Estimating transition probabilities in markov chain-based deterioration models for management of wastewater systems. **Journal of water resources planning and management**, 132(1):15–24, 2006.

[2] Flavia Barsotti, Yohann De Castro, Thibault Espinasse, and Paul Rochet. Estimating the transition matrix of a markov chain observed at random times. **Statistics & Probability Letters**, 94:98–105, 2014.

[3] David Blackwell. Conditional expectation and unbiased sequential estimation. **The Annals of Mathematical Statistics**, 18(1):105–110, 1947.

[4] Laurent E. Calvet and Adlai J. Fisher. How to forecast long-run volatility: Regime switching and the estimation of multifractal processes. **Journal of Financial Econometrics**, 2(1):49, 2004. doi: 10.1093/jjfinec/nbh003. URL `+http://dx.doi.org/10.1093/jjfinec/nbh003`.

[5] Muammer Catak. Application of markov chains on image enhancement. **Neural Computing and Applications**, 25(5):1119–1123, 2014. ISSN 1433-3058. doi: 10.1007/s00521-014-1591-3. URL `http://dx.doi.org/10.1007/s00521-014-1591-3`.

[6] Wai Ki Ching and Michael K Ng. **Markov chains**. Springer, 2006.

[7] David Roxbee Cox and David Victor Hinkley. **Theoretical statistics**. CRC Press, 1979.

[8] Richard D. Evans and Andrew G. Mueller. Industrial real estate cycles: Markov chain applications. **Journal of Real Estate Portfolio Management**, 22(1):75–90,

2016. URL https://colorado.idm.oclc.org/login?url=http://search.proquest.com.colorado.idm.oclc.org/docview/1792778406?accountid=14503.

[9] James D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. **Econometrica**, 57(2):357–384, 1989. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1912559.

[10] Yen-Ting Hu, Rudiger Kiesel, and William Perraudin. The estimation of transition matrices for sovereign credit ratings. **Journal of Banking & Finance**, 26(7):1383–1406, 2002.

[11] Leon Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. **Biometrika**, 12(1/2):134–139, 1918.

[12] Yongliang Jin and Amlan Mukherjee. Markov chain applications in modelling facility condition deterioration. **International Journal of Critical Infrastructures**, 10(2):93–112, 2014.

[13] S. E. Kuznetsov and V. I. Orlov. On unbiased estimation of a power of unknown matrix. In **3 rd All-Union school on Software and Algorithms for Multivariate Analysis**. Erevan, 1987. In Russian.

[14] Erich Leo Lehmann and Henry Scheffé. Completeness, similar regions, and unbiased estimation. ii. **Sankhyā: the Indian Journal of Statistics**, 15(3).

[15] Erich Leo Lehmann and Henry Scheffé. Completeness, similar regions, and unbiased estimation. i. **Sankhyā: the Indian Journal of Statistics**, 10(4):305–340, 1950.

[16] James D Nichols, John R Sauer, Kenneth H Pollock, and Jay B Hestbeck. Estimating transition probabilities for stage-based population projection matrices using capture-recapture data. **Ecology**, 73(1):306–312, 1992.

[17] José J Ortiz-García, Seósamh B Costello, and Martin S Snaith. Derivation of transition probability matrices for pavement deterioration modeling. **Journal of Transportation Engineering**, 132(2):141–161, 2006.

[18] Noelia Oses. Markov chain applications in the slot machine industry. **OR Insight**, 21(1): 9–21, 2008.

[19] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[20] Phillip E Pfeifer and Robert L Carraway. Modeling customer relationships as markov chains. **Journal of interactive marketing**, 14(2):43–55, 2000.

[21] C Radhakrishna Rao. Information and accuracy attainable in the estimation of statistical parameters. **Bull. Calcutta Math. Soc**, 37(3):81–91, 1945.

[22] George G Roussas. **A course in mathematical statistics**. Academic Press, 1997.

Appendix A

Additional Proofs

A.1 Proof of Fisher-Neyman Factorization Theorem

Theorem (Fisher-Neyman Factorization Theorem)**.** *Suppose the data $X = \{X_1, \ldots, X_k\}$ has probability distribution function $f_X(x_1, \ldots, x_k, \theta)$. Then a statistic $T(X)$ is sufficient for the parameter θ if and only if f_X factors into two components, $h(X)$ and $g_\theta(T, \theta)$, where h does not depend on θ, and g does not depend on X except as a function of T.*

Proof. First, if a statistic S is sufficient for the data X and parameter θ, then $f_{X|S}(x|s)$ does not depend on θ. So, we can write $f_X(x) = f_{X|S}(x|s)f_S(s)$, which factors as desired.

Now suppose f_X factors into $h(X)$ and $g_\theta(S, \theta)$, $f_X(x) = h(x)g_\theta(s, \theta)$.

$$\begin{aligned} f_{X|S}(x|s) &= \frac{f_{X,S}(x, s = S(x))}{f_S(s)} \\ &= \frac{h(x)g_\theta(s, \theta)}{f_S(s)} \end{aligned}$$

In the discrete case, we can compute $f_S(s)$ as $\sum_{\{y|S(y)=s\}} f_X(y) = \sum_{\{y|S(y)=s\}} h(y)g_\theta(s, \theta)$, and g_θ cancels out of the fraction. This computation is slightly more complicated in the continuous case, but the same key idea applies: $f_S(s)$ will contain a factor of g_θ, which then cancels from the numerator.

Let $S = (Y_1, ..., Y_m)$ and $X = (X_1, ..., X_n)$, where we assume $m < n$. Let $Y = (Y_1, ..., Y_n) = (u_1(X), ..., u_n(X))$, where u_i, $i > m$ are chosen such that $u = (u_1, ..., u_n)$ is invertible, and let $w = u^{-1}$. Now $f_Y(y) = f_X(w(y))|J(u)|$, where $J(u)$ is the Jacobian of u. Note J is independent of

θ. Then

$$\begin{aligned} f_Y(y) &= f_X(w(y))|J(u)| \\ &= h(w(y))g_\theta(s,\theta)|J(u)|. \end{aligned}$$

To recover $f_S(s)$ we will integrate out y_i for $i > m$. This will not change $g_\theta(s,\theta)$. Let $h^*(s)g_\theta(s,\theta)$ be the result of integrating over these y_i. Now we can say

$$\begin{aligned} f_{X|S}(x|s) &= \frac{f_{X,S}(x, s = S(x))}{f_S(s)} \\ &= \frac{h(x)g_\theta(s,\theta)}{h^*(s)g_\theta(s,\theta)} \\ &= \frac{h(x)}{h^*(s)}, \end{aligned}$$

which does not depend on θ. □

A.2 Proof of Rao-Blackwell Theorem

Theorem (Rao-Blackwell Theorem)**.** *Given an unbiased estimate for $g(\theta)$, $\varphi(X)$, and a sufficient statistic $T(X)$ of some parameter θ of the data X, the Rao-Blackwell estimator is $\hat{\varphi}(X) = E[\varphi(X)|T(X)]$. The Rao-Blackwell estimator is again unbiased for $g(\theta)$, and has variance no greater than the original estimator.*

Proof. The proof that the Rao-Blackwell estimate is an improved estimate follows from properties of conditional expectation.

First, given the expectations exist, $E[E[\varphi(X)|T]] = E[\varphi(X)]$, so $\hat{\varphi}(X) = E[\varphi(X)|T]$ is still unbiased.

Second, the variance of $\hat{\varphi}(X)$ will be less than the variance of $\varphi(X)$. We have $E[\varphi(X)] = g(\theta)$

and $\hat{\varphi}(X) = E[\varphi(X)|T]$. Then

$$\begin{aligned}
\mathrm{Var}(\varphi(X)) &= E[(\varphi(X) - g(\theta))^2] \\
&= E[((\varphi(X) - \hat{\varphi}(X)) + (\hat{\varphi}(X) - g(\theta)))^2] \\
&= E[(\varphi(X) - \hat{\varphi}(X))^2 + (\hat{\varphi}(X) - g(\theta))^2 + 2(\varphi(X) - \hat{\varphi}(X))(\hat{\varphi}(X) - g(\theta))]
\end{aligned}$$

We will show that $E[(\varphi(X) - \hat{\varphi}(X))(\hat{\varphi}(X) - g(\theta))] = 0$, and since $(\varphi(X) - \hat{\varphi}(X))^2$ is positive, we will have $\mathrm{Var}(\varphi(X)) \geq E[(\hat{\varphi}(X) - g(\theta))^2] = \mathrm{Var}(\hat{\varphi}(X))$ as desired.

$$\begin{aligned}
E[(\varphi(X) - \hat{\varphi}(X))(\hat{\varphi}(X) - g(\theta))] &= E[\varphi(X)\hat{\varphi}(X) - \hat{\varphi}(X)^2 - \varphi(X)g(\theta) + \hat{\varphi}(X)g(\theta)] \\
&= E[\varphi(X)\hat{\varphi}(X) - \hat{\varphi}(X)^2] - g(\theta)^2 + g(\theta)^2 \\
&= E[E[\varphi(X)\hat{\varphi}(X)|T]] - E[\hat{\varphi}(X)^2] \\
&= E[\hat{\varphi}(X)^2] - E[\hat{\varphi}(X)^2] \\
&= 0.
\end{aligned}$$

Finally, we must check that $\hat{\varphi}(X)$ is an observable estimate (ie, not depending on the parameter θ). This is guaranteed by the sufficiency of T. By definition, $f(X|T)$ does not depend on θ, so clearly $\hat{\varphi}(X) = E[\varphi(X)|T]$ does not depend on θ either. □

A.3 Proof of Lehmann-Scheffé Theorem

Theorem (Lehmann-Scheffé Theorem)**.** *Given the same hypotheses as in Rao-Blackwell, if we also assume $T(X)$ is a complete statistic for θ, $\hat{\varphi}(X)$ is the unique MVUE for $g(\theta)$.*

Proof. Let $\hat{\varphi}(X)$ be an unbiased estimate of $g(\theta)$ conditioned on the sufficient and complete statistic T. Let $\psi(X)$ be any other unbiased estimate of $g(\theta)$. Then $\hat{\psi}(X) = E[\psi(X)|T]$ is also unbiased (by Rao-Blackwell). Because T is complete, and $E[\hat{\varphi}(X) - \hat{\psi}(X)] = 0$, we must have $\hat{\varphi}(X) - \hat{\psi}(X) = 0$

almost everywhere. Finally, by Rao-Blackwell,

$$\begin{aligned}\mathrm{Var}(\psi(X)) &\geq \mathrm{Var}(\hat{\psi}(X)) \\ &= \mathrm{Var}(\hat{\varphi}(X)).\end{aligned}$$

So $\hat{\varphi}(X)$ has variance no greater than that of any other unbiased estimate of $g(\theta)$, and because T is complete, is the unique estimate with this property. □

A.4 Derivation of Cramér-Rao Bound

The Cramér-Rao bound is a lower bound on the variance of an estimate. Given an estimate $\varphi(X)$, with expectation $g(\theta)$, for a parameter θ, the Cramér-Rao bound is given by

$$\mathrm{Var}(\varphi(X)) \geq \frac{|g'(\theta)^2|}{I(\theta)},$$

where $I(\theta)$ is the Fisher information, defined as $I(\theta) = -E\left[\frac{\partial^2 l(x;\theta)}{\partial\theta^2}\right]$. $l(x:\theta)$ is the natural log of the likelihood function.

We wish to calculate the lower bound on the variance of $\varphi_t(X)$, with expectation μ^t, where the parameter θ is simply μ. So the numerator of the Cramer-Rao bound is $t^2\mu^{2t-t}$. Assuming our data consists of k observations x_i,

$$\begin{aligned}l(x:\theta) &= \log\left(\frac{1}{(2\pi\sigma^2)^{k/2}} e^{\frac{-\sum(x_i-\theta)^2}{2\sigma^2}}\right) \\ &= -\log((2\pi\sigma^2)^{k/2}) - \frac{\sum(x_i-\theta)^2}{2\sigma^2}.\end{aligned}$$

Then taking partial derivatives,

$$\begin{aligned}\frac{\partial^2 l(x:\theta)}{\partial\theta^2} &= \frac{\partial^2}{\partial\theta^2} - \log((2\pi\sigma^2)^{k/2}) - \frac{\sum(x_i-\theta)^2}{2\sigma^2} \\ &= \frac{\partial}{\partial\theta}\frac{2\sum(x_i-\theta)}{2\sigma^2} \\ &= \frac{\partial}{\partial\theta}\frac{\sum(x_i)-k\theta)}{\sigma^2} \\ &= \frac{k}{\sigma^2}.\end{aligned}$$

So the Cramér-Rao bound is $\frac{t^2\mu^{2t-t}}{k/\sigma^2} = \frac{t^2}{k}\mu^{2t-2}\sigma^2$.

Appendix B

Mathematica Code

B.1 Finding minimum variance by exhaustion

The following code blocks were used to compute the minimum variance for for different values of t, given $k = 3$ observations. The first block sets up bookkeeping for later steps (the order in which each coefficient is solved needed to be specified to keep Mathematic from failing).

In[1]:=

```
k=3;
list = Range[k] ;(* list from 1 to k *)
permutations = Permutations[list];
orders = Length[permutations];
σ = 1;(* rescaling *)

dep[1] = {};
dep[2] = Array[co,2];
dep[3] = Array[co,2];
dep[4] = Array[co,3];
dep[5] = Array[co,3];
dep[6] = {co[1],co[2],co[3],co[5]};
dep[7] = {co[1],co[2],co[3],co[5]};
```

```
dep[8] = {co[1],co[2],co[3],co[5],co[7]};
dep[9] = {co[1],co[2],co[3],co[5],co[7]};
dep[10] = {co[1],co[2],co[3],co[5],co[7],co[10]};
dep[11] = {co[1],co[2],co[3],co[5],co[7],co[10]};
dep[12] = {co[1],co[2],co[3],co[5],co[7],co[10],co[13]};
dep[13] = {co[1],co[2],co[3],co[5],co[7],co[10],co[13]};
```

The next block uses the functions from the previous block to compute the expectation of a general degree t polynomial, and eliminates coefficients by setting that expectation equal to μ^t. To speed up the calculation, we let $\mu = 1$, which still gives a general solution, as previous calculations had shown that for the values of t used, the coefficients that minimize variance do not depend on μ or σ^2.

Once we have constrained the coefficients to ensure the estimate is unbiased, we compute the variance and solve for the coefficients which minimize the variance, saving the value of the minimum variance.

```
In[2]:= t = 6;
types = IntegerPartitions[t,k];(* list partitions to use in exponents *)
len = Length[types]; (* number of terms *)
(* create general degree t polynomial *)
For[j = 1,j ≤ len, j++,
  (* loop through the partitions of t, and create monomials  *)
  exponents = PadRight[types[[j]],k];
  mono[j] = 0;
  For[jj = 1, jj ≤ orders, jj++,
    (* for each monomial, take every order of the observations and
```

```
    raise to appropriate exponents*)
    type[jj] = Product[x[permutations[[jj]][[ii]]]^(exponents[[ii]]),
                {ii,1,k}];
    mono[j] = mono[j] + type[jj];
  ];
];
poly = Sum[co[i]mono[i],{i,1,len}];
(* iteratively take expectations to get mean and variance *)
expt = poly;
For[i = 1, i ≤ k, i++,
  expt = Expectation[expt,Distributed[x[i],NormalDistribution[μ,σ]]]
];
var = (poly-μ^t)^2;
For[i = 1, i ≤ k, i++,
  var = Expectation[var,Distributed[x[i],NormalDistribution[μ,σ]]]
];

coeffs = Join[ConstantArray[0,t],{1}]; (* coefficients of powers of μ *)
(* solve for expt = μ^t, eliminating variables listed in dep first *)
var =
   var/.Solve[Thread[CoefficientList[expt,μ]==coeffs],dep[t]]/.{μ -> 0};
(* assume μ = 0 to simplify calculations *)
indep = Complement[Array[co,len],dep[t]]; (* uneliminated variables *)
zero = ConstantArray[0,Length[indep]];
variance = var/.Solve[Thread[Thread[D[var,{indep}]]==zero],indep];
```

Finally, an estimate of the form (1.1) is built, and the variance is calculated. We then set the variance equal to the minimum variance found earlier, and solve for the coefficients $a1$ and $a2$. For values of t less than 12, $a1 = (1+\sqrt{3})/6, a2 = -1/3$ achieves the minimum variance calculated previously. For $t = 12$, no solution was found.

```
In[3]:=   d0 = (x1+x2+x3)/3;
          a3 = -a1-a2;


          xxx1 = (d0 + i(a1 x1 + a2 x2 + a3 x3))^t ;
          xxx2 = (d0 + i(a1 x1 + a3 x2 + a2 x3))^t;
          xxx3 = (d0 + i(a2 x1 + a1 x2 + a3 x3))^t;
          xxx4 = (d0 + i(a3 x1 + a1 x2 + a2 x3))^t;
          xxx5 = (d0 + i(a3 x1 + a2 x2 + a1 x3))^t;
          xxx6 = (d0 + i(a2 x1 + a3 x2 + a1 x3))^t;
          meanpoly = ComplexExpand[(Re[(xxx1+xxx2+xxx3+xxx4+xxx5+xxx6)]/6)];
          varpoly = ComplexExpand[(meanpoly)^2];  (* assuming μ = 0 *)
          var1 = Expectation[Expectation[Expectation[varpoly,
                   Distributed[x1,NormalDistribution[0,1]]],
                     Distributed[x2,NormalDistribution[0,1]]],
                       Distributed[x3,NormalDistribution[0,1]]];
          Solve[var1==variance[[1]] && a1^2+a2^2+a3^2==1/3, {a1,a2}, Reals]
```

B.2 Verifying Formula (2.1) for various t

The following code was run to verify (2.1). For each value of t, the code verifies that the formula holds for all k. The code was run for t up to $t = 32$. The computation for $t = 32$ took over 47 hours and no further computations were completed.

In the following code, for fixed t, we first expand the $S_1^a S_2^b$ terms. If we start with $k = t$, we can figure out the binomial coefficients of each type of term, then add a coefficient in terms of k.

$$P = \frac{t!(k-3)!!}{(k+t-3)!} \sum_{i=0}^{\lfloor t/2 \rfloor} \frac{(-1)^i (k+2t-2i-4)!! S_1^{2t-2i} S_2^i}{(t-2i)!(2i)!!}.$$

So we need to expand terms of the form $S_1^{2t-2i} S_2^i$. poly[i] will represent $S_1^{t-2i} S_2^i$. Since the X_i are iid, we do not need to distinguish between terms with the same exponents, but different variables. We only need to count how many times each type will occur.

```
In[4]:= For[t = 10, t ≤ 20, t++,
        startTime = AbsoluteTime[];
        types = IntegerPartitions[t];
        len = Length[types];
        s1 = Sum[x[i], {i, 1, t}];
        s2 = Sum[x[i]^2, {i, 1, t}];
        For[j = 1, j ≤ len, j++,
          (* loop through the partitions of t, compute a coefficient based
          on the number and repetitions of exponents, and form monomials *)
          (* ie, x1^2 x2 x3 gets coefficient k(k-1)(k-2)/2! *)
          exponents=types[[j]];
          type[j] = Product[x[ii]^(exponents[[ii]]),{ii,1,Length[exponents]}];
          multiplicities = BinCounts[exponents];
          coeff[j] = Product[k-ii,{ii,0,Length[exponents]-1}]/
                  Product[multiplicities[[ii]]!,{ii,1,Length[multiplicities]}];
        ];
        For[i = 0, i ≤ t/2, i++,
          (* for each type of term (based only on the numbers appearing as
```

```
    exponents), we want to take the coefficient from expanding the s1s2
    term, and add the coefficient calculated earlier in terms of k, then
    sum those terms *)
    terms = s1^(t-2i) s2^i;
    poly[i] =
      Sum[Coefficient[terms,type[j]]coeff[j]type[j],{j,1,len}]/(k^(t-i));
  ];
  P = Sum[poly[i] (-1)^i t! (k-3)!!(k+2t-2i-4)!! / ((t-2i)!(2i)!!(k+t-3)!),{i,0,Floor[t/2]}];
  (* poly[i] now takes the place of s1^(t-2i)s2^i *)
  expt = P;
  For[i = 1, i ≤ t, i++,
    expt = Expectation[expt,Distributed[x[i],NormalDistribution[μ,σ]]]
    (* iteratively take expectations *)
  ];
  expt = FullSimplify[expt];
  endTime = AbsoluteTime[];
  Print[{t,expt==μ^t,endTime-startTime}];
  (* if expt == mu^t returns True, we have an unbiased estimate! *)
];
```

B.3 Verification of variance formula for various t

The following code calculates the variance for various t (valid for all k), and compares it to the conjectured variance. The conjectured variance was confirmed for t up to $t = 13$.

For fixed t, we compute the variance of $\hat{P}$ in terms of k. We know the distributions of S_1 and S_2 in terms of k. We scale S_2 so that we can take the expection of a χ^2 variable (this is simpler

than telling Mathematica how to scale S_2 itself).

```
In[5]:= For[t = 2, t < 7, t++,

  startTime = AbsoluteTime[];
  P = Sum[Binomial[t,2i] (2i-1)!!(k-3)!!(-1)^i / (k+2i-3)!! s1^(t-2i) (σ^2 s2/k)^i, {i, 0, Floor[t/2]}];
  var = (P-μ^t)^2;

  expt = Expectation[Expectation[var,

          Distributed[s1,NormalDistribution[μ,σ/√k]]],

            Distributed[s2,ChiSquareDistribution[k-1]]];
  guess = Sum[i!(k-3)!!(k+2i-4)!!Binomial[t,i]^2 μ^(2t-2i) σ^(2i) / ((k+i-3)! k^i), {i, 1, t}];
  endTime=AbsoluteTime[];

  Print[{t,FullSimplify[expt==guess],endTime-startTime}];

];
```

B.4 Verification of Asymptotic Efficiency of MVUE for various t

The following code compares the variance of (2.2) to the Cramer-Rao lower bound. For t up to $t = 15$, the code verified that the estimator is asymptotically efficient. The computation for $t = 15$ took over 12 hours and no further computations were completed.

```
In[6]:= For[t = 2, t < 16, t++,

  startTime = AbsoluteTime[];
  P = Sum[Binomial[t,2i] (2i-1)!!(k-3)!!(-1)^i / (k+2i-3)!! s1^(t-2i) (σ^2 s2/k)^i, {i, 0, Floor[t/2]}];
  var = (P-μ^t)^2;

  expt = Expectation[Expectation[var,

          Distributed[s1,NormalDistribution[μ,σ/√k]]],
```

```
                Distributed[s2,ChiSquareDistribution[k-1]]];
    expt = FullSimplify[expt];
    CramerRao = (t^2 μ^(2t-2) σ^2/k);
    lim = Limit[expt/CramerRao,k→Infinity];
    endTime = AbsoluteTime[];
    Print[{t,lim==1,endTime-startTime}];
];
```

B.5 Verification of Asymptotic Efficiency of Recommended Estimate for various t

The following code compares the variance of (4.2) to the Cramer-Rao lower bound. For t up to $t = 6$, the code verified that the estimator is asymptotically efficient. Further verification has not yet been attempted. For $t < 6$, each verification took 4 or fewer seconds; for $t = 6$, the computation took 40 seconds.

We compute the variance following the expansion of (4.2) as in Section 4.1. The variance is given through expanding (4.1), and considering which types of cross-terms we will encounter. Given

$$\left(\sum_{1 \le j_1 < j_2 \le k} \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{2i} \left(\frac{-1}{2k} \right)^i \bar{X}^{t-2i} (X_{h_1} - X_{h_2})^{2i} \right)^2,$$

we will have three types of cross-terms:

$$\sum_{j=0}^{\lfloor t/2 \rfloor} \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{2i} \binom{t}{2j} (-1)^{i+j} \frac{1}{2k}^{i+j} \bar{X}^{t-2i-2j} (X_{h_1} - X_{h_2})^{2i} (X_{h_3} - X_{h_4})^{2j}.$$

We must consider the number of terms where $h_1 = h_3$ and $h_2 = h_4$, the number of terms where one of h_1 or h_2 matches h_3 and h_4, and the number of terms where all 4 observations are different. First, $X_{h_1} - X_{h_2}$ is equivalent to $X_{h_2} - X_{h_1}$, since this term is always raised to an even power. So, there are $\binom{k}{2}$ ways to come up with the term $(X_{h_1} - X_{h_2})^{2i+2j}$. There are $\binom{k}{3}$ ways to choose the

three observations showing up in the next term, and $\binom{3}{1} = 3$ ways to choose an observation to put in the second product, then $\binom{2}{1} = 2$ ways to select an observation from the first product to repeat in the second. Finally, there are $\binom{k}{4}\binom{4}{2} = 6\binom{k}{4}$ ways to come up with the last type of term. A bit of algebra verifies that these add up to the proper number of total terms ($\binom{k}{2}^2$), as well.

```
In[7]:= For[t = 2, t < 7, t++,

  startTime = AbsoluteTime[];
  P = Sum[Sum[Binomial[t,2i] Binomial[t,2j] (-1)^(i+j) 4 s1^(2t-2i-2j) /
      ((2k)^(i+j) k^2 (k-1)^2)
     (Binomial[k,2](x[1]-x[2])^(2i+2j)
      + 6 Binomial[k,3](x[1]-x[2])^(2i)(x[1]-x[3])^(2j)
      + 6 Binomial[k,4](x[1]-x[2])^(2i)(x[3]-x[4])^(2j)),
    {j,0,Floor[t/2]}],{i,0,Floor[t/2]}];
  var = (P-μ^(2t));
  expt = Expectation[Expectation[var,
           Distributed[s1,NormalDistribution[μ,σ/√k]]],
             Distributed[{x[1],x[2],x[3],x[4]},
               MultinormalDistribution[{μ, μ, μ, μ},
                 σ^2 IdentityMatrix[4]]]];
  expt = Simplify[expt];
  CramerRao = (t^2 μ^(2t-2)σ^2/k);
  lim = Limit[expt/CramerRao,k→Infinity];
  endTime = AbsoluteTime[];
  Print[{t,lim,endTime-startTime}];
];
```

B.6 Construction of estimate (3.8)

The following code was used to constructed the matrix estimate (3.8). The constructions works for any t, given $k = 3$ observations of 2×2 matrices.

The first block sets up initial conditions, and is used to initialize all of the following matrix code (Appendices B.6-B.8).

```
In[8]:= k = 3;
        s1 = Sum[(y[i])/k, {i, 1, k}];
        s2 = Sum[(y[i]-s1)^2/k, {i, 1, k}];
        (* 2x2 matrix *)
        x[1] = ({{x111, x112},
                 {x121, x122}});
        x[2] = ({{x211, x212},
                 {x221, x222}});
        x[3] = ({{x311, x312},
                 {x321, x322}});
        m = ({{m1, m2},
              {m3, m4}});
        id = ({{1, 0},
               {0, 1}});
```

Next, for fixed t, a list of the monomials appearing in the scalar MVUE is generated. For each term in this list, the coefficient of the term that appears in the MVUE is saved, then a matrix product is built for every ordering of observations in the term. These products are averaged, and the appropriate coefficient is added.

```
In[9]:=  (* implementation of new formula *)

         startTime = AbsoluteTime[];

         P = 0;

         terms = MonomialList[
                   Sum[(-1)^i (2i-1)!! Binomial[t,2i] / Product[k+2j-3, {j, 1, i}] s1^(t-2i) s2^i,
                     {i, 0, Floor[t/2]}]];

           (* list of monomial terms in P *)

         For[i = 1, i ≤ Length[terms], i++,

           term = terms[[i]];

           x1exp = Exponent[term,y[1]];

           x2exp = Exponent[term,y[2]];

           x3exp = Exponent[term,y[3]];

           exponents={x1exp,x2exp,x3exp};

             (* list of the exponents appearing in this type of term *)

           type = y[1]^x1exp y[2]^x2exp y[3]^x3exp;

             (* strip the coefficient from the term *)

           coefficient = term/type; (* only the coefficient remains *)

           list = {};

           For[ii = 1, ii ≤ Length[exponents], ii++,

             list = Join[list, ConstantArray[ii, exponents[[ii]]]];

           ];

           orders = Permutations[list]; (* Get permutations of the term *)

           divisor = Length[orders]; (* count number of permutations *)

           permutations = 0;

           For[j = 1, j ≤ Length[orders], j++,

             order = orders[[j]];
```

```
    current = id;
    For[ii = 1, ii ≤ t, ii++,
      current = current.x[order[[ii]]];
      (* add each observation to the product in order *)
    ];
    permutations += current; (* sum all permutations *)
  ];
  term = coefficient permutations /(divisor);
    (* add coefficient and average *)
  P += term;
];
```

B.7 Timing of Matrix Estimates

The following code generates random matrices with which to construct the estimates discussed in this thesis. We use $k = 3$, and compare the time it tkaes to construct the estimates for t up to $t = 14$. For each t, the estimates are constructed using 10 different random matrices, and the average, maximum, and minimum times are compared.

First, we initialize as in Appendix B.6, Next, for each t, we loop over 10 random realizations of the observations. For each loop, first the estimate (3.8) is constructed, then (4.2) is constructed.

```
In[10]:= For[t = 3, t < 15, t++,
  newTime = ConstantArray[0,10];
  oldTime = ConstantArray[0,10];
  For[iter = 1, iter ≤ 10, iter++,
    x111 = RandomReal[];
```

```
x112 = RandomReal[];

x121 = RandomReal[];

x122 = RandomReal[];

x211 = RandomReal[];

x212 = RandomReal[];

x221 = RandomReal[];

x222 = RandomReal[];

x311 = RandomReal[];

x312 = RandomReal[];

x321 = RandomReal[];

x322 = RandomReal[];

(* implementation of new formula *)

startTime = AbsoluteTime[];


  ...construction of estimate as in B.6...


endTime = AbsoluteTime[];

newTime[[iter]] = endTime-startTime;

(* implementation of recommended formula *)

startTime = AbsoluteTime[];

d0 = (x[1]+x[2]+x[3])/3;

a1 = 1/Sqrt[2k];

a2 = -1/Sqrt[2k];

xxx1 = MatrixPower[(d0 + i(a1 x[1] + a2 x[2])),t];

xxx2 = MatrixPower[(d0 + i(a1 x[1] + a2 x[3])),t];

xxx3 = MatrixPower[(d0 + i(a1 x[2] + a2 x[3])),t];
```

```
    P2 = ComplexExpand[(Re[(xxx1+xxx2+xxx3)]/3)];
    endTime = AbsoluteTime[];
    oldTime[[iter]] = endTime-startTime;
  ];
  Print[{t, Mean[newTime], Max[newTime], Min[newTime]}];
  Print[{t, Mean[oldTime], Max[oldTime], Min[oldTime]}];
];
```

B.8 Theoretical Comparison of Matrix Estimates

The initial set up and construction of the estimate (3.8) is the same as in Appendix B.6. Once this estimate has been constructed, the following code is used to construct the recommended estimate and compute the covariance matrices of the each estimate's entries.

```
In[11]:=  (* implementation of recommended formula *)
d0 = (x[1]+x[2]+x[3])/3;
a1 = 1/Sqrt[2k];
a2 = -1/Sqrt[2k];
xxx1 = MatrixPower[(d0 + i(a1 x[1] + a2 x[2])),t];
xxx2 = MatrixPower[(d0 + i(a1 x[1] + a2 x[3])),t];
xxx3 = MatrixPower[(d0 + i(a1 x[2] + a2 x[3])),t];
P2 = ComplexExpand[(Re[(xxx1+xxx2+xxx3)]/3)];
  (* calculate covariance matrix based on new formula *)
startTime = AbsoluteTime[];
vector1 = Flatten[(P-MatrixPower[m,t])];
varvec1 = Partition[vector1,1].Partition[vector1,4];
```

```
way1 = Table[
       Expectation[Expectation[Expectation[Expand[varvec1[[i,j]]],
         Distributed[{x111,x112,x121,x122},
           MultinormalDistribution[{m1,m2,m3,m4},
             σ^2 IdentityMatrix[4]]]],
         Distributed[{x211,x212,x221,x222},
           MultinormalDistribution[{m1,m2,m3,m4},
              σ^2 IdentityMatrix[4]]]],
         Distributed[{x311,x312,x321,x322},
           MultinormalDistribution[{m1,m2,m3,m4},
             σ^2 IdentityMatrix[4]]]],
       {i, 1, 4}, {j, 1, 4}];
endTime = AbsoluteTime[];
Print[{t,endTime-startTime}];


  (* calculate covariance matrix based on recommended formula *)
startTime = AbsoluteTime[];
vector2 = Flatten[(P2-MatrixPower[m,t])];
varvec2 = Partition[vector2,1].Partition[vector2,4];
way2 = Table[
       Expectation[Expectation[Expectation[Expand[varvec2[[i,j]]],
         Distributed[{x111,x112,x121,x122},
           MultinormalDistribution[{m1,m2,m3,m4},
             σ^2 IdentityMatrix[4]]]],
         Distributed[{x211,x212,x221,x222},
           MultinormalDistribution[{m1,m2,m3,m4},
```

```
                σ^2 IdentityMatrix[4]]]],
          Distributed[{x311,x312,x321,x322},
            MultinormalDistribution[{m1,m2,m3,m4},
              σ^2 IdentityMatrix[4]]]],
        {i, 1, 4}, {j, 1, 4}];
endTime = AbsoluteTime[];
Print[{t,endTime-startTime}];


  (* compare new vs old *)
diff = way2 - way1;
Eigenvalues[diff]
```