DATA COMPRESSION IN STATISTICAL LEARNING BY MEANS OF QUANTIZED RANDOM PROJECTION

by

Glenn Hui

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Statistical Science

Committee:

_____	Dr. Martin Slawski, Dissertation Director
_____	Dr. Anand Vidyashankar, Committee Member
_____	Dr. Daniel Carr, Committee Member
_____	Dr. Emanuel Ben-David, Committee Member
_____	Dr. Anand Vidyashankar, Department Chair
_____	Dr. Kenneth S. Ball, Dean, The Volgenau School of Engineering
Date: _____	Summer Semester 2020 George Mason University Fairfax, VA

Data Compression in Statistical Learning by Means of Quantized Random Projection

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Glenn Hui
Master of Science
London School of Economics, 2005
Bachelor of Science
University of Toronto, 2004

Director: Dr. Martin Slawski, Professor
Department of Statistics

Summer Semester 2020
George Mason University
Fairfax, VA

Dedication

To T.K.H.

Acknowledgments

I would like to thank the following people who made this possible: Dr. Martin Slawski, my advisor, who helped me grow as a researcher and stuck with me when I was really stuck. Dr. Anand Vidyashankar, Dr. Daniel Carr, and Dr. Emanuel Ben-David for insight and advice that helped make my dissertation the best it can be. Dr. William Rosenberger for his guidance and faith in me. Verronica Mitchell, Carroll Barbour, and Liz Quigley for their help in keeping the departmental engines running. Various friends for proof reading then humoring me when I try to explain what they just read. And last but certainly not least, my parents for their love and support.

Table of Contents

Page

List of Figures . . . viii

Abstract . . . xi

1 Introduction . . . 1

2 Chapter 2 . . . 4

2.1 Random Projection Background . . . 6

2.1.1 Preservation of pairwise distances and inner products . . . 6

2.1.2 Quantization and other variants of random projection . . . 7

2.2 Random Projection Theoretical Foundation . . . 9

2.2.1 Algebraic Formulation of Random Projection . . . 9

2.2.2 Unbiasedness of Distance Estimator . . . 10

2.3 Estimators of Pairwise Distances and Inner Products . . . 13

2.3.1 Applications and Related Work . . . 16

3 Chapter 3 . . . 18

3.1 Quantization Methods Defined for Scalars . . . 19

3.1.1 Deterministic Quantization . . . 19

3.1.2 Stochastic Quantization . . . 20

3.2 Comparison of Estimators of ρ after Deterministic and Stochastic Quantization . . . 21

3.2.1 Estimators of ρ . . . 21

3.2.2 Expectation and variance of 1-bit discrete quantized estimator $\hat{\rho}_{q_M}$ . . . 22

3.2.3 $\hat{\rho}_{q_S}$, Stochastic Quantization estimator . . . 30

3.2.4 Analysis of $\hat{\rho_S}$, estimator after Stochastic 4-level / 2-bit Quantization . . . 33

3.2.5 General m-level estimator $\hat{\rho_S}$ . . . 35

3.2.6 MSE . . . 38

3.3 Other estimators of ρ in the presence of quantization . . . 41

3.4 Application to Spectral Clustering . . . 44

3.4.1 Quantized experiments . . . 44

3.5 Discussion . . . 45

4 Chapter 4 . . . 47
4.1 Bounds for Estimators of ρ after Random Projection . . . 48
4.1.1 Variance of $g(\hat{\rho}_{lin})$ . . . 48
4.1.2 Expectation and bias of $g(\hat{\rho}_{lin})$ . . . 53
4.1.3 Expectation of $\hat{\rho}_{mult}$ . . . 55
4.2 Bounds for Gaussian Similarity Measure . . . 60
4.2.1 Expectation of $g(\hat{\rho}_{mult})$ . . . 60
4.2.2 Variance of $g(\hat{\rho}_{mult})$ . . . 61
4.3 Bounds for Gaussian measure $g(\hat{\rho})$ for quantized $\hat{\rho}$ . . . 65
4.3.1 Delta method for expectation $g(\hat{\rho}_{q_M})$, 1-bit Quantized MLE . . . 65
4.3.2 Delta method / Taylor expansion for expectation $g(\hat{\rho}_{q_S})$, Stochastic Quantization estimator . . . 66
4.3.3 Bounds for $g(\hat{\rho}_{lin})$, general b bit quantized estimator . . . 68
4.4 Discussion . . . 72
5 Chapter 5 . . . 74
5.1 Spectral Clustering Simulation Experiments . . . 74
5.1.1 Simulation Setting Details . . . 75
5.1.2 Initial Results: Clustering Accuracy against ρ and k . . . 76
5.1.3 Linear versus Multiplicative Estimator . . . 77
5.1.4 Variance of clustering accuracy . . . 83
5.1.5 Ratios and Differences of Similarities . . . 83
5.2 Spectral Clustering and Random Projection Algorithm Parameters . . . 86
5.2.1 Gaussian kernel parameter σ . . . 86
5.2.2 Known correlation between clusters ρ . . . 87
5.2.3 Reduced dimensionality k . . . 87
5.2.4 AR(1) simulation results . . . 90
5.3 Real data experiments . . . 92
5.3.1 MNIST dataset . . . 92
5.3.2 Columbia University Image Library (COIL-20) dataset . . . 95
5.4 Discussion . . . 96
A Appendix 1 . . . 97
A.1 Section 2.1, Johnson-Lindenstrauss for inner products . . . 97
A.2 Section 2.2.2 Alternative Proof . . . 98
A.3 Section 3.2 Proof using Isserlis' theorem . . . 102

A.4 Section 3.2 Lemmas . . . 103
A.5 Delta Method using Taylor Expansion Detailed . . . 104
A.6 Section 3.2.3 discssion on selection of b . . . 105
A.7 Section 3.2.4 . . . 108
A.7.1 Section 4.1, full calculation using Taylor expansions for $E(\hat{\rho})$ . . . 110
B Appendix 2 . . . 112
B.1 Spectral Clustering Experiment Details . . . 112
B.1.1 Spectral Clustering Implementation . . . 113
B.1.2 Experiment 1 Details . . . 115
B.1.3 Experiment 1 Results . . . 116
B.1.4 Experiment 2 Details . . . 118
B.1.5 Experiment 2 Results . . . 120
Bibliography . . . 121

List of Figures

Figure Page

2.1 Random Projection conceptualized. 7

3.1 Simulated $\hat{\rho}_{q_M}$ results using 1-bit MLE demonstrating that both are $O\left(\frac{1}{k}\right)$. 29

3.2 Variance of 2-bit $\hat{\rho_S}$, levels = $\{-4.0, -0.5, 0.5, 4.0\}$ corresponding to a = 0.5, b = 4.0. Theoretical and simulated values are plotted against each other, with the simulated values confirming that the variance matches (3.24). . . . 36

3.3 Theoretical MSE of $\hat{\rho}$ by ρ (uncompressed data's correlation), demonstrating that 1-bit Discrete Quantization has a lower MSE than 1-bit Stochastic Quantization, and still better than 2- and 3-bit Q_s for $\rho <$ 0.6. 40

3.4 Classification Accuracy of QRP + SC on MNIST, averaged results over all pairs. We can see that quantized performance rapidly converges to that of full precision with only 2 or 3 bits. 46

4.1 Calculated and simulated results for Taylor expansion remainder terms of Variance of $\hat{\rho}$. In these plots we plot both the actual values, along with plotted curves with regression lines, verifying that (left) $\mathrm{E}(\hat{\rho} - \rho)^3 = O\left(\frac{1}{k^2}\right)$, (right) Bias^2 $(\hat{\rho})$ is $O\left(\frac{1}{k}\right)$. 51

4.2 log |bias| vs ρ, separate models for each k. In this plot we demonstrate that bias $\rho = O\left(\frac{1}{k}\right)$; we plot log values to aid clarity. 59

4.3 Plot of log $\mathrm{Var}(g(\rho_{mult}))$ against ρ, simulated and theoretical; we plot log values so that differences are clearer. The close fit shows that the true values of $\mathrm{Var}(g(\hat{\rho}_{mult}))$ helps confirm our calculated $O\left(\frac{1}{k}\right)$. 64

4.4 log MSE of $\hat{\rho}$, quantized linear estimator. We vary levels of $b \cdot k = 384$. Theoretical upper bound as calculated above in Eq. 4.19. While this plot is mostly to compare simulated values and theoretical upper bounds, we also note that MSE improves rapidly as b increases from 1 to 3, where it starts to stablize. 73

5.1 Clustering Accuracy, Regime 1, n = 1000, replicates = 100, $\sigma = 1$. These two plots are the initial results using the linear estimator $\hat{\rho}_{lin}$, demonstrating the relationship of Accuracy with ρ (true, original correlation) and k (reduced dimensionality). 77
5.2 Bias and variance of $g(\hat{\rho})$ for both linear and multiplicative estimators . . . 80
5.3 Clustering accuracy, Regime 1, n = 2000, replicates = 100. Besides the relationship between Accuracy and both ρ and k, this plot shows that the two estimators $\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$ perform similarly, with $\hat{\rho}_{lin}$ performing slightly and consistently better. 82
5.4 Variance of accuracy, Regime 1, n = 1000, replicates = 100, $\hat{\rho}_{lin}$. The goal of these plots is to highlight how clustering accuracy varies more when accuracy is in the "middle" range of around 0.65 to 0.85 (noting that Accuracy is between 0.5 and 1.0 for two clusters). 83
5.5 Variance of accuracy, Regime 1, n = 1000, replicates = 100, $\hat{\rho}_{lin}$. These are two of the plots we earlier combined in Fig. 5.3, with standard error bars to demonstrate how much accuracy can vary. 84
5.6 Difference and Ratio of g($\hat{\rho_w}$) / g($\hat{\rho_b}$), theoretical and simulated results. The aim of the ratio and difference plots is to show how separation in $g(\hat{\rho}) = \exp((\hat{\rho} - 1)/\sigma^2)$ varies with k and (true, pre-compression) ρ. The expected relationship with ρ is clear. We can also see, as k increases, the simulated plots follow the theoretical lines more closely. 85
5.7 Difference of g(ρ_w) - g(ρ_b) and ratio of g(ρ_w) / g(ρ_b) against σ, theoretical and simulation results. 87
5.8 Relationship between Clustering Accuracy and Difference g(ρ_w) - g(ρ_b), across σ. Each of the four panels fixes a different value of ρ. Ultimately the plots show us that the difference in similarity measures does not tell the entire story. 88
5.9 Plots of theoretical $g(\rho)$. Each plot has a different value of σ; within each plot are standard error bars for different levels of k. The goal of these plots is to show how rapidly the compressed data approaches the true values as k increases. In particular we can see that the intersection with $g(\rho)$ approaches $\rho = 0$ when $\sigma = 1$. 89
5.10 Accuracy vs ρ, regime 2 (AR(1) ρ within blocks), for various σ. 90

5.11 Variance of accuracy vs ρ, regime 2 (AR(1) ρ within blocks), for selected values of k. 91
5.12 MNIST Accuracy vs $\rho_{within} - \rho_{between}$, varying k. 20 replicates, $\sigma = 1.0$. . 93
5.13 MNIST Accuracy vs $\rho_{within} - \rho_{between}$, varying σ. 50 replicates, $k = 150$. . . 94
5.14 Accuracy vs k, Spectral clustering. RP vs PCA vs full data, no quantization. 95
A.1 Values of b corresponding to various α and $n \cdot k$. We can see that even for extreme values, setting $b = 7$ is sufficiently high. 106
B.1 RP and PCA, $k \leq 300$ 117
B.2 RP, $k \geq 400$ 117
B.3 Experiment 1 Results 117
B.4 Experiment 2 Results 120

Abstract

DATA COMPRESSION IN STATISTICAL LEARNING BY MEANS OF QUANTIZED RANDOM PROJECTION

Glenn Hui, PhD

George Mason University, 2020

Dissertation Director: Dr. Martin Slawski

This dissertation explores advancements in random projection, a method of dimensionality reduction that reduces the number of variables in a dataset via multiplication with a smaller, randomly generated matrix. Compared to other dimensionality reduction methods, random projection has very fast operation time and a small memory footprint, at the cost of potentially more information loss. This dissertation investigates the effects of quantization on random projection, particularly in conjunction with machine learning algorithms. We demonstrate that in many settings the amount information loss can be managed to be of minimal practical consequence.

Beyond establishing an effective method of data compression for machine learning, this dissertation provides formulae and code that allow future researchers, or anyone who seeks to reduce their data under certain circumstances, to determine an optimal compression scheme. We also feel that the structure of our proofs can be generalized to suit them a variety of machine learning algorithms, or quantization methods.

The first two chapters of this dissertation introduce the topic and provide background information, respectively. Chapter 2 goes into significant algebraic detail to provide context for understanding the following chapters. Chapter 3 investigates quantization, with two different methods of quantization compared both individually and when combined with random projection. It focuses on proofs on bounds of mean squared errors. Chapter 4 focuses on random projection in conjunction with spectral clustering, an algorithm whose strengths align well with those of random projection. Again the focus is on analyzing bounds. Finally, chapter 5 explores these algorithms through experiments. Two simulation experiments further explore the technical details of spectral clustering and random projection, before we close with some real data experiments that compare random projection to principal components analysis.

Chapter 1: Introduction

Introduction

As computing power and storage capabilities have grown, so has the amount of data we store: large datasets now abound in all corners of industry, research, and government. With large data sets come associated difficulties. In particular, as dimensionality – the number of variables in a dataset, or columns if the data are thought of as a table or matrix – grows, datasets become difficult to interpret, statistical analysis is burdened by the increasing sample size required for statistical significance, and computation time explodes. Even core operations required for data analysis, such as matrix inversion, become intractable, making techniques for dealing with massive datasets increasingly in-demand.

A common tool employed when dealing with large datasets is data compression, which is the encoding of data to use less storage space, and is discussed in contexts ranging from imaging, big data, signal processing, and communications. When compressing data, there are several considerations needed: the amount of time taken to compress the data (and reconstruct it if required), the amount of compression (savings), and the amount of information loss. All of these trade-offs will be addressed throughout this dissertation.

There are many approaches to data compression, but this dissertation is primarily interested in dimensionality reduction, a class of methods that aim to reduce the number of variables stored or used in a dataset. One such method is random projection (RP, or RPs for random projections), which takes an original dataset of dimension d and reduces it to dimension k — perhaps orders of magnitude lower than d – by multiplying the data in matrix form by a randomly generated matrix.

This dissertation investigates advancements in and applications of random projection,

primarily around quantized random projection (QRP). Quantization is a process of transforming continuous data to discrete; in the simplest case, any continuous scalar can be reduced to one bit, either positive or negative. Quantized random projection, then, can be thought of as element-wise quantization of randomly projected data. We focus on two sub-types of quantization, and combining with RP: deterministic quantization and stochastic quantization. Where deterministic quantization assigns a discrete value based solely on where the original value is, stochastic uniform quantization assigns a discrete value with probability equal to the distance of the real point from each neighboring threshold or "cut" point. We discuss this in detail in Chapter 3.

Overview of Contributions

This dissertation explores the theory and practice of using machine learning algorithms on data that has been compressed via quantized random projection. To be precise, we focus on applying RP to a data set before using the compressed data as input to spectral clustering. Spectral clustering is a state-of-the-art clustering method that creates a graph out of a dataset and uses the eigenvectors of this graph's similarity matrix to perform clustering [1], which is what random projection aims to preserve. In many contexts spectral clustering is among the best performing methods, has several performance guarantees, and is non-parametric and thus broadly applicable. Specific applications with promising experimental results include speech separation [2] and biotechnology [3]. Finally, we briefly explore the impact of quantized random projection on other machine learning algorithms.

The setting of our experiment is a high-dimensional data set with n observations and d variables, $X \in \mathbb{R}^{n\times d}$. For clarity, by "data compressed via quantized random projection", we mean that the input for the machine learning algorithms shall not be the original data $X \in \mathbb{R}^{n\times d}$ but compressed data $Q \in (\mathcal{M}^{\pm})^{n\times d}$, where Q is the element-wise quantized version of $Z = XR$, as described in section 3.1, and $\mathcal{M}^{\pm}$ is a set of codes representing discrete partitions of the real line. This concept is defined in detail in (3.1).

We can divide our work into theoretical and applied areas. In terms of theory, we perform analysis on how quantization affects estimators, expected values, and variances. We calculate bounds for bias and variance, the first step of which is to assess the bias and variance of the Gaussian similarity $s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ using the various "quantized" estimators described above. (We note, again, that the estimators themselves are not quantized, thus the quotation marks; we use the term "quantized estimators" to refer to estimators based on quantized compressed data.) We define Gaussian similarity function formally here, for some known tuning parameter σ:

$$s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \quad x, x' \mapsto \exp(-\|x - x'\|^2/2\sigma^2) \tag{1.1}$$

As part of this exploration, we shall examine the ideal usage of storage space, or more specifically the relationship between b bits and k compressed dimensions. To illustrate this idea, let us assume that for our purposes we have 256 bits to store each observation of some data set after compression: we could store that as, say, 256 columns of 1-bit data ($k = 256$, $b = 1$), or 128 columns of 2-bit data ($k = 128$, $b = 2$), and so on.

We confirm this analysis with experiments and simulations, most of these presented in line with relevant work. The final chapter also contains some applied experiments involving spectral clustering on real data that has been compressed via QRP.

Chapter 2: Random Projection and Dimensionality Reduction

Dimensionality reduction is an increasingly important field, and within that field random projection has been gathering interest due to its speed, especially in comparison to principal components analysis (PCA). In this chapter we present some fundamental proofs that later work will build upon and establish background theory.

Chapter outline

We begin with theoretical definitions of random projections and proofs on pairwise distances to establish the context of our work. We move on to theoretical foundation and comparisons with PCA.

Notation Table

$I(\cdot)$	Indicator function
$\equiv$	Equality by definition
X	A script capital letter generally refers to a matrix.
x or x'	A script lowercase letter generally refers to a vector. Shorthand x, x' indicates an arbitrary pair of observations from a dataset X.
x_i	The i^{th} scalar element of vector x.
x_{ij}	The i, j scalar element of matrix X.
$x_{i,:}$	The i^{th} row vector of matrix X.
$x_{:,j}$	The j^{th} column vector of matrix X.
$\|x\|_q$	L_q-norm of a vector $x \in \mathbb{R}^d$, i.e. $\|x\|_q = \left(\sum_{i=1}^{d} x^q\right)^{1/q}$. $\|x\|$ implicitly means the L_2-norm if subscript q is missing
$\langle x, x' \rangle$	Usual Euclidean inner product of vectors $x, x' \in \mathbb{R}^n$:
$x \sim N(\mu, \Sigma)$	Random vector x follows a Gaussian (normal) distribution with mean vector μ and covariance matrix Σ
$\underset{r}{argmin} f(\cdot, r)$	The value of r that minimizes $f(\cdot, r)$.
$\underset{r}{argmax} f(\cdot, r)$	The value of r that maximizes $f(\cdot, r)$.
$f^{(j)}(\cdot)$	The j^{th} derivative of $f(\cdot)$
i.i.d.	Independent and identically distributed
w.p.	With probability
$\log(x)$	Natural logarithm of x
$\text{Bias}_1(\hat{\rho})$	First order Taylor expansion approximation of $\text{Bias}(\hat{\rho})$, where $\hat{\rho}$ is an estimator of some observed value ρ.
$\text{Var}_1(\hat{\rho})$	First order Taylor expansion approximation of $\text{Var}(\hat{\rho})$
$k = \Omega(f(n))$	k is asymptotically lower bounded by $f(n)$
$Q_D(\cdot)$	Discrete quantization as a function of some real-valued input
$Q_S(\cdot)$	Stochastic quantization as a function of some real-valued input

2.1 Random Projection Background

We begin with a precise definition of random projection and the setting we work in. Given n observations with d variables, represented as $X \in \mathbb{R}^{n \times d}$, the rows of X are the observations and the columns of X are the variables. We refer to each observation i of X as row vector x_i, and an arbitrary pair of observations as x, x'. Our goal is to compress this data with minimal loss in pairwise distances $\langle x, x' \rangle$ for each pair of observations x, x' in the data set. Random projection creates a compressed data set $Z \in \mathbb{R}^{n \times k}$ by multiplying X by a random $d \times k$ matrix $R \in \mathbb{R}^{d \times k}$: that is $Z = XR$ [4]. (We refer to observations of Z, corresponding to arbitrary original observations x, x', as z, z'.) "Random projection" can in fact be used to describe various implementations, such as forming $Z = RX$ where $X \in \mathbb{R}^{d \times n}$ has observations in columns and features in rows [5], but we will always apply RP as $Z = XR$.

The elements of $R = (r_{ij}), 1 \leq i \leq d, 1 \leq j \leq k$, are independently and identically distributed (i.i.d.) random variables. A common case – and the one we focus on in our research – is to let each r_{ij} be a standard normal random variable. Unless otherwise noted, henceforth RP will be assumed to be standard normal RP – or alternatively properly scaled, i.e., $r_{ij} \sim N(0, \frac{1}{k})$, where k is the reduced dimension. The scaling need not be done at this stage, but doing so makes later formulae cleaner. Then $Z = (z_{ij}), 1 \leq i \leq n, 1 \leq j \leq k$. We use the convention that x is an arbitrary observation (row) of the original data X, and z is the corresponding observation of the compressed data Z. We treat the data x as fixed, and work with properties of the projected data z conditional on x. We illustrate this concept via a diagram of block matrices, Fig. 2.1.

2.1.1 Preservation of pairwise distances and inner products

Random projection preserves approximate Euclidean distance between any two data points (rows of X) x and x': we can bound the difference between pairwise distances of compressed

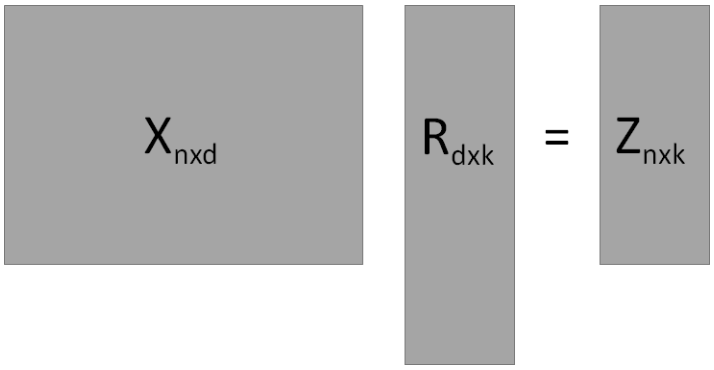


Figure 2.1: Random Projection conceptualized.

data points z and z' compared to the true distances between the original data points. Formally, this preservation of squared distances is guaranteed by the Johnson-Lindenstrauss lemma [6], which, in the context of RP, gives us that for any error factor $\epsilon \in (0, 1)$, there is some dimension $k = \Omega\left(\frac{\log(n)}{\epsilon^2}\right)$, such that

$$(1-\epsilon)\|x-x'\|^2 \leq \|z-z'\|^2 \leq (1+\epsilon)\|x-x'\|^2$$

This preservation extends to inner products as well, that is:

$$\langle x, x' \rangle - \epsilon \cdot \|x\|\|x'\| \leq \langle z, z' \rangle \leq \langle x, x' \rangle + \epsilon \cdot \|x\|\|x'\|$$

Appendix A.1 contains a derivation of this statement.

It follows that random projection preserves both pairwise distances and inner products, a characteristic we shall make implicit use of frequently throughout this dissertation.

2.1.2 Quantization and other variants of random projection

Quantization is a procedure that partitions the real line into disjoint intervals ("bins"); each bin is associated with a code, typically some real value within its associated bin. Quantization, like random projection, aims to save on storage space, and is often seen in

applications where bit rate is at a premium such as signal and image processing. Combining these two methods by quantizing the randomly projected data, element-wise, allows for further storage savings, particularly for applications where throughput is a primary concern: thus Quantized Random Projection (QRP) [7]. It is not uncommon to apply quantization with other methods, such as in hashing [8, 9] or compressed sensing [10].

We explore two methods of quantization: Deterministic Quantization (specifically Lloyd-Max quantization[11], which we may also refer to as simply "quantization" when context is clear) as well as Stochastic Quantization (a method that essentially flips a weighted coin to choose which bin to assign to each value) [12]. Our overall process is the same for either case: letting $Q_u(\cdot)$ be the quantization function, quantized randomly projected matrix Q consists of element-wise quantization of randomly projected matrix Z, i.e.

$$Q = (q_{ij}) = Q_u(\mathbf{x}_i^T \mathbf{r}_j), i \in \{1, 2, ..., n\}, j \in \{1, ..., k\}$$

We illustrate the concept in matrix form below.

$$Z = \begin{pmatrix} \mathbf{x}_1^T\mathbf{r}_1 & \mathbf{x}_1^T\mathbf{r}_2 & ... & \mathbf{x}_1^T\mathbf{r}_k \\ \mathbf{x}_2^T\mathbf{r}_1 & \mathbf{x}_2^T\mathbf{r}_2 & ... & \mathbf{x}_2^T\mathbf{r}_k \\ ... & & & \\ \mathbf{x}_n^T\mathbf{r}_1 & \mathbf{x}_n^T\mathbf{r}_2 & ... & \mathbf{x}_n^T\mathbf{r}_k \end{pmatrix} \qquad Q = \begin{pmatrix} Q_u(\mathbf{x}_1^T\mathbf{r}_1) & Q_u(\mathbf{x}_1^T\mathbf{r}_2) & ... & Q_u(\mathbf{x}_1^T\mathbf{r}_k) \\ Q_u(\mathbf{x}_2^T\mathbf{r}_1) & Q_u(\mathbf{x}_2^T\mathbf{r}_2) & ... & Q_u(\mathbf{x}_2^T\mathbf{r}_k) \\ ... & & & \\ Q_u(\mathbf{x}_n^T\mathbf{r}_1) & Q_u(\mathbf{x}_n^T\mathbf{r}_2) & ... & Q_u(\mathbf{x}_n^T\mathbf{r}_k) \end{pmatrix}$$

Comparison to structured random projection

It is worth making the distinction between quantized random projection and various "structured" random projections. Achlioptas [13] has shown that sign random projections – that is, random projections where the projection matrix simply takes two values, positive and negative (+1 / -1) with equal probability – still provide consistent estimation of distances between points. He also proposed "sparse" random projections, in which each element of

the projection matrix takes on three values instead of two: -1, 0, +1, with respective probability 1/6, 2/3, 1/6. These structured RPs are distinct from QRP in that structured RPs can be thought of as quantizing entries of the projection matrix, whereas QRP quantizes after projection is completed.

2.2 Random Projection Theoretical Foundation

The next sections establish foundational proofs involving random projection (RP). We occasionally compare results to those of principal component analysis (PCA, or PCs for principal components) to help contextualize.

2.2.1 Algebraic Formulation of Random Projection

We begin by establishing some syntax. As mentioned above, given n observations of d-dimensional Euclidean data $X \in \mathbb{R}^{n\times d}$, random projection reduces dimensionality to k-dimensional Euclidean data $Z \in \mathbb{R}^{n\times k}$ by multiplying by a random $d \times k$ matrix $R \in \mathbb{R}^{d\times k}$ – that is, $Z = XR$. First note that $Z = (z_{ij}) = (\frac{1}{\sqrt{k}} x_{i,:} r_{:,j})$. We can also represent X and Z by their respective rows (data points), i.e.

$$X = \begin{bmatrix} x_{1,:}^T \\ x_{2,:}^T \\ \vdots \\ x_{n,.}^T \end{bmatrix} \text{ and } Z = \begin{bmatrix} z_{1,:}^T \\ z_{2,:}^T \\ \vdots \\ z_{n,:}^T \end{bmatrix}, \qquad z_{i,:}^T = [z_{i1}, z_{i2}, \ldots, z_{ik}]$$

Having presented the basic formulations explicitly, we drop the above notation going forward and return to a simpler notation: we indicate a vector simply by a letter, ie z instead of $z_{i,:}$, and denote a pair of arbitrary observations (vectors) as z and z'. Using this notation,

Euclidean distance is

$$d^2(x, x') = \|x - x'\|^2 = \|x\|^2 + \|x'\|^2 - 2\langle x, x' \rangle.$$

When dealing with unit norms (i.e., $\|x\| = 1$ for all rows / observations x of X), this reduces to

$$d^2(x, x') = \|x - x'\|^2 = 2(1 - \langle x, x' \rangle) = 2(1 - \rho),$$

where ρ is the cosine similarity of (cosine of the angle between) x and x'. This reduces to the inner product $\langle x, x' \rangle$ for unit length vectors.

2.2.2 Unbiasedness of Distance Estimator

We focus our work on Gaussian (normal) Random Projections, where each element of $R = (r_{ij})$, $1 \leq i \leq d, 1 \leq j \leq k$ is a $N(0, 1)$ random variable. Such RPs provide unbiased estimators for $d^2(x, x') = \|x - x'\|^2$, which we prove below.

Proposition 2.2.1. *Estimating distance after random projection is unbiased,*
i.e. $E[d^2(z, z')] = d^2(x, x')$.

Proof. We show that $E[d^2(z, z')] = d^2(x, x')$. By definition,

$$d^2(z, z') = \|z - z'\|^2 = \|z\|^2 + \|z'\|^2 - 2\langle z, z' \rangle \text{ , and}$$

$$E[\|z\|^2] = E\left[\sum_{i=1}^{k} z_i^2\right] = E\left[\sum_{i=1}^{k} \left(\sum_{j=1}^{d} x_j r_{ji}\right)^2 / k\right] \tag{2.1}$$

Since each $r_{ji} \sim N(0, 1)$, we have

$$E(x_j \cdot r_{ji}) = x_j \cdot E(r_{ji}) = 0, \text{ and}$$

$$E(x_j^2 \cdot r_{ji}^2) = x_j^2 \cdot E(r_{ji}^2) = x_j^2$$

So each term in the summation in (2.1) above, $E(z_i^2) = E(x'^2 \cdot r_{ji}^2) = x'^2$

$$E[\|z\|^2] = \sum_{i=1}^{k} x'^2 = \|x\|^2$$

Similarly, $E\|z'\|^2 = \|x'\|^2$. It now remains to show that $E\langle z, z'\rangle = \langle x, x'\rangle$.

$$\begin{aligned} E\langle z, z'\rangle &= E\left[\sum_{i=1}^{k} z_i z_i'\right] = E\left[\sum_{i=1}^{k}\left(\sum_{j=1}^{d} x_j r_{ji} \sum_{j=1}^{d} x_j' r_{ji}\right)\right] \\ &= E\left[r_{11}^2(x_1 x_1') + r_{22}^2(x_2 x_2') + \cdots + r_{dd}^2(x_d x_d')\right] \\ &= E\left[(x_1 x_1') + (x_2 x_2') + \cdots + (x_d x_d')\right] \\ &= \sum_{i=1}^{d} x_i x_i' = \langle x, x'\rangle \end{aligned}$$

Thus $E\langle z, z'\rangle = \langle x, x'\rangle$, and so in all $E\left[d^2(z, z')\right] = d^2(x, x')$. □

Remark

We used three properties of random projections in the above proof: that for all $i, j \in 1, 2, \ldots d$, 1) $E[r_{ij}] = 0$, 2) $\mathrm{Var}(r_{ij}) = 1$, and 3) for all k, l such that $!\,(k = i, l = j), Cov(r_{ij}, r_{kl}) = 0$. Thus, the above proof applies to any such random projection.

Proposition 2.2.2. *The estimator for distance is also consistent.*

Proof. To prove consistency it suffices to show that $d^2(z, z') \to d^2(x, x')$ in probability, i.e.

$$\forall \epsilon, \lim_{k\to\infty} P\left[\mid d^2(z, z') - d^2(x, x') \mid \geq \epsilon\right] = 0 \tag{2.2}$$

We first show that the $\mathrm{Var}(d^2(z, z'))$ is $O\left(\frac{1}{k}\right)$. Note that $d^2(z, z') = \mid z - z' \mid^2$, and that $z - z'$ is a normal random vector. To see this, note that

$$z - z' = \frac{1}{k}\sum_{j=1}^{d} x_j r_{ij} - \frac{1}{k}\sum_{j=1}^{d} x'_j r_{ij}$$

$$= \frac{1}{k}\sum_{j=1}^{d}(x_j - x'_j) r_{ij}$$

which is a linear combination of normal random variables, and thus is normally distributed itself. It follows that each element of $z - z'$ has finite variance, and thus

$$\sigma_d^2 \equiv Var(d^2(z, z')) = \text{Var}\left(\frac{1}{k}\sum_{j=1}^{d}(x_j - x'_j) r_{ij}\right)$$

$$= O\left(\frac{1}{k}\right)$$

We now turn to Chebyshev's Inequality, which states that, for any random variable X with finite mean μ and finite variance σ^2, for all $\epsilon > 0$

$$P(\mid X - \mu \mid \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$$

And so in our case,

$$P(\mid d^2(z, z') - d^2(x, x') \mid \geq \epsilon) \leq \frac{\sigma_d^2}{\epsilon^2}$$

$$= \frac{1}{\epsilon^2} \cdot O\left(\frac{1}{k}\right)$$

Taking the limit as $k \to \infty$, we have

$$\lim_{k\to\infty} P\left[\mid d^2(z, z') - d^2(x, x') \mid \geq \epsilon\right] = 0$$

Which satisfies (2.2). □

As an alternative proof, one can show that $\|z\|, \|z'\|,$ and $\langle z, z' \rangle$ are consistent estimators of $\|x\|, \|x'\|,$ and $\langle x, x' \rangle$ respectively. Since $d^2(z, z') = \|z\|^2 + \|z'\|^2 - 2\langle z, z' \rangle$, it follows that $d^2(z, z')$ is a consistent estimator for $\|x\|^2 + \|x_i'\|^2 - 2\langle x, x' \rangle = d^2(x, x')$.

2.3 Estimators of Pairwise Distances and Inner Products

Many machine learning and multivariate statistics algorithms use the input data via pairwise distances or inner products between observations. Examples include k-means clustering, spectral clustering, linear regression, Principal Components Analysis (PCA), Linear Discriminant Analysis, and Canonical Correlation Analysis. Random projection's preservation of pairwise distances and inner products thus allows us to apply these algorithms on compressed data with bounded error. Estimation of the original data X's pairwise distances or inner products using the compressed data Z is thus a key aspect of applying RP, and our research.

In this context our topic of interest becomes the trade off of bit rate against accuracy of pairwise distance estimation, instead of the usual paradigm of having random data and (say) reducing the variance of a sample. We note that we are thus treating the input data X as observed constants; we are only concerned with the randomization of the compression (random projection).

Example Uses for Pairwise Distance Estimation

The spectral clustering algorithm is a prime example application of pairwise estimation. In spectral clustering, actual data points are discarded entirely for an n-node graph – represented as an $n \times n$ similarity matrix, where n is the number of observations – made up of pairwise distances, such as Gaussian similarity $s(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$.

Another example is linear regression, which – while normally thought of as a function of inner products between variables – can be represented as a function of the inner products

between observations, i.e. via the dual form. In the dual form, we solve for the coefficient $\alpha = (XX^T)^{-1}y$ [14], which can be transformed to the familiar coefficient estimator $\beta = X^T\alpha$. Calculating the dual form of linear regression on projected data $Z = XR$ involves ZZ^T (instead of Z^TZ, as the primal form does), i.e. the pairwise distances preserved by random projection. These sorts of examples inform our focus on estimators of ρ.

Other examples include K-nearest neighbors classification [15], Support Vector Machines [16], and PCA itself [17].

Estimators of ρ after random projection

Perhaps the most straightforward estimator of $\langle x, x' \rangle$ is $\langle z, z' \rangle$. This is of course also a natural estimator of ρ, which we call the "linear" estimator $\widehat{\rho}_{lin} = \langle z, z' \rangle$. This estimator is unbiased (i.e., $E(\langle z, z' \rangle) = \rho)$) and consistent, as we demonstrated in the previous section. It is also simple to calculate, and has variance $\frac{1+\rho^2}{k}$ [7].

Now, note that each pair of points $(z_l, z'_l), l \in \{1, 2, ..., k\}$ have distribution

$$(z_l, z'_l)^T \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{\|x_l\|^2}{k} & \frac{\langle x_l, x'_l \rangle}{k} \\ \frac{\langle x_l, x'_l \rangle}{k} & \frac{\|x'_l\|^2}{k} \end{pmatrix}\right).$$

Then, assuming the data are 0-mean, $\langle z, z' \rangle$ is the sample covariance of the pairs (z_{il}, z_{jl}), and thus is the MLE and uniform minimum variance unbiased estimator (UMVUE) of $\langle x, x' \rangle$ in the case where we do not have individual norms [19].

We can improve upon the linear estimator above if we assume the norms of the original data observations $\|x\|$ have been calculated and stored; in this context, we can assume that $\|x\| = 1$ for all $i \in \{1, \cdots, n\}$. Doing so is efficient: calculation of the $\|x\|$ can be done as the data are read in, and storage is only of the order of $O(n)$. Given the $\|x\|$, inner product and normalized inner product are equivalent. We shall henceforth refer to normalized inner product as $\rho = \langle x, x' \rangle / \|x\| \|x'\|$, and work in the unit norm context. Note also that in the

unit norm case, inner product and L^2 norm are also equivalent without even needing the norms, as then $\|x - x'\|^2 = 2(1 - \rho)$.

The papers [19] and [7] discuss two other closed form estimators that depend only on norms: the "simple margin" estimator $\hat{\rho}_{add}$ and "normalized" estimator $\hat{\rho}_{mult}$:

$$(i)\ \hat{\rho}_{add} = 1 - \frac{1}{2}\|z - z'\|^2 \qquad (ii)\ \hat{\rho}_{mult} = \frac{\langle z, z' \rangle}{\|z\|\|z'\|}$$

The former is unbiased and has variance $2(1-\rho)^2/k$, lower than $\widehat{\rho}_{lin}$ when $\rho > 2-\sqrt{3} \approx 0.27$. The latter is asymptotically unbiased with variance $(1 - \rho)^2/k$, uniformly lower than $\widehat{\rho}_{lin}$ (but usually higher than that of $\hat{\rho}_{add}$).

Non-quantized MLE

Finally, Li [19] suggests the maximum likelihood estimator (MLE), which uses the fact that each pair of z, z' are bivariate Gaussian with correlation ρ. The MLE then results as

$$\hat{\rho}_{MLE} = \underset{r}{argmin} \left(-\frac{1}{2}\log(1 - r^2) - \frac{1}{2}\frac{1}{1 - r^2}(\|z\|^2 + \|z'\|^2 - 2r\langle z, z' \rangle) \right)$$

One can show further that $\hat{\rho}_{MLE}$ is a solution of the following cubic equation in ρ:

$$\rho^3 - \rho^2\langle z, z' \rangle + \rho(-1 + \|z\|^2 + \|z'\|^2) - \langle z, z' \rangle = 0$$

The MLE has lower asymptotic variance than the other estimators, but its non-closed form solution makes it unwieldy.

Our first step involving estimators is to analyze the effects of the various estimators (in quantized context) on the Gaussian similarity mentioned above, $s(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$, plugging in our various estimators into the distance formula. In particular, we note that by the invariance property of the MLE, the MLE of any function of an estimator is simply that estimator plugged into the function. We conduct the analysis on bias and

variance, and perform simulations to see how an algorithm (specifically, spectral clustering) using this distance performs.

2.3.1 Applications and Related Work

Over the past several years, researchers have looked into applications of machine learning algorithms to data compressed by random projection. In particular, the comparison to Principal Components Analysis is a natural and common one; both methods can be thought of as a matrix multiplication that reduces the original data in matrix form $X \in \mathbb{R}^{n \times d}$ to some $Z \in \mathbb{R}^{n \times k}$. We discuss PCA and its relationship to RP further in subsection "Random Projection Algebraic Background" below.

In [20] and [21], Dasgupta compares RP to PCA in the context of learning mixtures of Gaussians, in particular discussing "c-separation", the distance between the centers (means) of Gaussians to be compared. He theorizes that the "spherical" effects of RP – that is, data have lower eccentricity after RP – would make them efficient under certain contexts, and finds that combining RP with expectation-maximization (EM) works in Optical Character Recognition (OCR) and simulations.

In [22], Boutsidis et al. examine sign random projection in conjunction with k-means clustering. They establish bounds for performance: given an error bound $\epsilon \in (0, 1/3)$, k-means clustering can be performed on the compressed data while preserving clustering structure within a factor of $2+\epsilon$. (In other words, the objective function of the approximated clustering is within $2 + \epsilon$ of the optimal clustering's objective function.)

They test both accuracy and run-time performance via a facial recognition task and find that RP performs only slightly worse in terms of misclassification rates than several competing methods, with performance often orders of magnitude faster. (It is worth noting here that we deemed run-time optimization outside of the scope of this dissertation.)

In [23], Bingham and Mannila discuss RP as applied to image processing (using images of nature) and text data. They compare PCA and discrete cosine transform to both Gaussian and Sparse RP, over ranges of k ranging from 1 to 800. They find that RP does not introduce

significant distortion to their data, and performs significantly better than discrete cosine transform which is a computationally fast algorithm like RP. Furthermore, they add noise to their data and find that RP is relatively robust, suggesting that it could be useful for noise reduction. Generally, they note that RP mitigates the curse of dimensionality and may be useful for cases where other methods of dimensionality reduction may be computationally infeasible. A similar survey found that random projection performs fairly well in general contexts [24].

Several surveys have been done in which RP is compared with PCA, including [25], in which they run both methods before nearest neighbor clustering and find comparable results, and [26], which looks at several other dimensionality reduction methods as well for cancer image classification.

Other applications that have been explored with promising results so far include facial recognition [27] via neural networks [28, 29], nearest neighbor classification [26], hyperspectral imaging [30, 31], cybersecurity (specifically anomalous behavior detection [32, 33]), image compression [34], and privacy for distributed computing [35, 36].

Chapter 3: Quantization Methods

In this chapter we explore quantization and its application to random projection. Quantization is a procedure that partitions the real line into disjoint intervals ("bins"); in the simplest case, any continuous scalar can be reduced to one bit, either positive or negative. Generally, each bin is associated with a code, typically some real value within its associated bin. (This is sometimes referred to as "scalar quantization", as it is quantizing a single number on the real line to a discrete subset of the real line. We note here that all of the quantization we do in this dissertation is on a scalar; if we discuss quantization of a matrix or vector, we mean performing scalar quantization on each element independently.) Quantization, like random projection, aims to save on storage space, and is often seen in applications where bit rate is at a premium such as signal and image processing. Combining these two methods for further storage savings can then be referred to as Quantized Random Projection (QRP) and has become a topic of some study [7].

Chapter Outline

This chapter is divided in to three parts. We begin with detailed definitions and background theory of quantization methods, then compare theoretical bounds for the Mean Squared Error (MSE) of these methods. We then explore the relationship between quantization and random projection. We close with practical experiments performing Spectral Clustering on data compressed via Quantized Random Projection that combine all the above theory. Throughout the chapter we will intersperse small simulations that demonstrate the theory.

3.1 Quantization Methods Defined for Scalars

In this section we define two methods of quantization, which we dub Deterministic Quantization (DQ, or $Q_D(\cdot)$ when referring to a function) and Stochastic Quantization (SQ or $Q_S(\cdot)$). (Recall that we denote the general quantization function $Q_u(\cdot)$.) At a high level, our methodology for quantized random projection is the same for each method: we begin with data $X \in \mathbb{R}^{n\times d}$, with n observations and d variables, then randomly project it to generate $Z = XR$, $Z \in \mathbb{R}^{n\times k}, R \in \mathbb{R}^{d\times k}$. We then perform element-wise quantization: letting $q_{ij} = Q_u(z_{ij})$ for each (i, j), we denote the quantized, compressed data set as $Q = (q_{ij}), 1 \leq i \leq n, 1 \leq j \leq k$.

3.1.1 Deterministic Quantization

We begin with a formal definition of deterministic quantization. Scalar deterministic quantization can be defined as a function Q_D from $\mathbb{R}$ to a set of codes (often called quantization alphabet) $\mathcal{M}^{\pm} = -\mathcal{M} \cup \mathcal{M}$, where $\mathcal{M} = \{\mu_i\}, i \in \{1, 2, ..., 2^{b-1}\}$. Each code μ_i represents one of the ordered, non-overlapping bins defined by its thresholds $[t_i, t_{i+1})$; for our work we assume that $\mu_i \in [t_{i-1}, t_i)$. Note that in our work the codes are symmetric around 0, as the projected data to be quantized are 0-mean. Equipped with those quantities, we are in position to define the map Q_D:

$$Q_D : \mathbb{R} \to \mathcal{M}^{\pm} \equiv -\mathcal{M} \cup \mathcal{M}, \quad z \mapsto Q_D(z) = \text{sign}(z) \sum_{s=1}^{2^{b-1}} \mu_s I(|z| \in [t_{s-1}, t_s)) \tag{3.1}$$

Quantized random projection then can be thought of as element-wise quantization of the randomly projected data Z, where $Z = XR$: letting $q_{ij} = Q_D(z_{ij})$ for each (i, j), we denote the quantized, compressed data set as $Q = (q_{ij}), 1 \leq i \leq n, 1 \leq j \leq k$. We may also represent this concept as $Q = Q_D(Z)$.

Unless otherwise noted, deterministic quantization in our research will be performed

according to the well-known Lloyd-Max algorithm [37, 38].

3.1.2 Stochastic Quantization

Stochastic Quantization, sometimes known as dithering or Stochastic Uniform Quantization in other data compression contexts [12, 39], is a probabilistic method of quantization that essentially flips a weighted coin to decide which of the two nearest cutpoints z is assigned to. Stochastic quantization eliminates bias but, as we will demonstrate, tends to have a larger variance. In this chapter we compare these methods' estimation of ρ theoretically, then verify with simulation. We define stochastic quantization $Q_S(z)$ formally here, as described in [39]. Let m be the number of levels (or cut-points) to quantize to. (If we are compressing down to b bits, the number of levels $m = 2^b$.) Since our randomly projected data will always be 0-mean, we can define the set of levels $\mathcal{B} = B_1, B_2, ..., B_m$ symmetrically about 0, so that $B_1 = -B_m$, $B_2 = -B_{m-1}$, etc. This is analagous to letting μ_i be the discrete values in an alphabet $\mathcal{M}$, except in SQ the representative values are also cut points. Then, for the two cut points B_i, B_{i+1} such that $z \in [B_i, B_{i+1})$, $Q_S(z)$ takes on the value of either with probability equal to the relative distance z is from each cut point. That is,

$$Q_S(z) = \begin{cases} B_i, \text{ w.p. } \mid \frac{z - B_{i+1}}{B_{i+1} - B_i} \mid \\ \\ B_{i+1}, \text{ w.p. } \mid \frac{z - B_i}{B_{i+1} - B_i} \mid \end{cases}$$

As an example, let us say we are performing 2-level stochastic quantization, with our levels at say $B_1 = -8, B_2 = 12$. If we have a data point $z = 6$, then we would have

$$Q_S(6) = \begin{cases} -8, \text{ w.p. } \mid 6 - 12 \mid /20 = 0.3 \\ \\ 12, \text{ w.p. } \mid 6 - (-8) \mid /20 = 0.7 \end{cases}$$

3.2 Comparison of Estimators of ρ after Deterministic and Stochastic Quantization

In this section we compare Deterministic Quantization and Stochastic Quantization in the context of estimating ρ, the inner product (or cosine similarity) of two vectors z, z' (i.e., $\rho = \langle z, z' \rangle$).

Given two data points z, z', we aim to estimate their cosine similarity ρ using the quantized (compressed) values $\langle q, q' \rangle$. This chapter compares two estimators of ρ, one for each of Discrete Quantization and Stochastic Quantization. We establish that the mean squared error (MSE) of DQ is lower than that of SQ when using the same number of bits, and thus preferable in most cases where the slight bias of DQ is not crucial.

As mentioned above, quantization is a method of transforming continuous data to discrete. In this section we discuss this in combination random projection, which can be thought of as element-wise quantization of randomly projected data. Specifically we combine each of DQ and SQ with RP.

Where quantization assigns a discrete value based solely on where the original value is, stochastic quantization assigns a discrete value to a neighboring value with probability proportional to the distance of the real point from each cut point.

3.2.1 Estimators of ρ

To estimate $\rho = \langle z, z' \rangle$ using SQ, we define

$$\hat{\rho}_{q_S} = \frac{1}{k} \sum_{l=1}^{k} q_l \cdot q'_l$$

Meanwhile, our estimator under 1-bit deterministic quantization is the MLE: $\hat{\rho}_{q_M}$:

$$\hat{\rho}_{q_M} = \cos\left(\pi\left(1 - \hat{P}_{sign}\right)\right) = \cos\left(\pi\left(1 - \frac{1}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right)$$

Where $P_{sign} = P(q_l = q'_l) = P\Big(\text{sign}(z_l) = \text{sign}(z'_l)\Big)$ is the probability of collision, and empirical collision probability $\hat{P}_{sign} = \frac{1}{k}\sum_{l=1}^{k} I(q_{il} = q_{jl})$ is our estimator thereof. k is, as usual, the reduced dimension of our random projection. We now proceed to establish bounds on the estimation error of these estimators.

3.2.2 Expectation and variance of 1-bit discrete quantized estimator $\hat{\rho}_{q_M}$

Theorem 3.2.1. *Bias* $(\hat{\rho}_{q_M}) \leq \frac{\rho}{2k}\left(\frac{\pi^2}{4} - \arcsin^2(\rho)\right) = O\left(\frac{1}{k}\right)$

Proof. We calculate the expectation of $\hat{\rho}$ and show that $\mathrm{E}[\hat{\rho}]$ is $\rho + O\left(\frac{1}{k}\right)$. Our first step is simplifying using $\cos(a+b) = (\cos a \cos b) + (\sin a \sin b)$. We will eventually use the delta method to bound the expectation of the simplified version below.

$$E(\hat{\rho}_{q_M}) = E\left[\cos\left(\pi\left(1 - \frac{1}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right)\right] \tag{3.2}$$

$$= E\left[\cos\pi \cdot \cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right] - E\left[(\sin\pi)\cdot\left(\sin\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right], \tag{3.3}$$

$$= E\left[-1 \cdot \cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right] - E\left[0 \cdot \left(\sin\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right], \tag{3.4}$$

$$= -E\left[\cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right] \tag{3.5}$$

Lemma 3.2.2. $E\left[\sum_l^k I(q_l = q'_l)\right] = k \cdot \left(\frac{1}{\pi}\arcsin(\rho) + \frac{1}{2}\right)$. *Note that for convenience we define* $S \equiv \sum_l^k I(q_l = q'_l)$, *so that we have* $E[S] = k \cdot \left(\frac{1}{\pi}\arcsin(\rho) + \frac{1}{2}\right)$.

Proof.

$$E\left[\sum_{l}^{k} I(q_l = q'_l)\right] = k \cdot P(q_l = q'_l) = k \cdot \left(\frac{1}{\pi}\arcsin(\rho) + \frac{1}{2}\right) \tag{3.6}$$

Where the last term can be calculated by integrating a bivariate normal pdf from 0 to ∞ across both dimensions [40], and noting that each $P(q_l = q'_l)$ is equal for all l. $\square$

Now we define a function f,

$$f : \mathbb{R} \to \mathbb{R}, \quad r \mapsto \cos\left(\frac{\pi}{k} \cdot r\right) \tag{3.7}$$

so that $\mathrm{E}[\hat{\rho}_{q_M}] = -E[f(S)]$, via Eq. 3.5. We can now differentiate with respect to $S = \sum_{l=1}^{k} I(q_l = q'_l)$ for the next step's Taylor expansion. For convenience in this section we also define $\mathrm{E}[S] = \mu$. We then have:

$$\frac{\partial f(S)}{\partial S} = -\frac{\pi}{k}\sin(\frac{\pi}{k}S), \qquad \frac{\partial^2 f(S)}{\partial S^2} = -\frac{\pi^2}{k^2}\cos(\frac{\pi}{k}S),$$

$$\frac{\partial f(\mu)}{\partial S} = -\frac{\pi}{k}\sqrt{1-\rho^2} \qquad \frac{\partial^2 f(\mu)}{\partial S^2} = -\rho$$

We now take the Taylor expansion around μ to solve for $\mathrm{E}(\hat{\rho})$. (For details on our application of the delta method using Taylor expansions, see Appendix A.5.)

$$E(\hat{\rho}) \equiv E[f(S)] = f(\mu) + E\left[\frac{f''(\tilde{S})}{2} \cdot (S-\mu)^2\right] \tag{3.8}$$

Where $\tilde{S} \in \left[S = [\sum_l^k I(q_l = q'_l)], k\rho\right]$. We now calculate the first order term and variance term in Equation (3.8) separately:

Lemma 3.2.3. *The first order term* $f(\mu) = -\cos(\frac{\pi}{k}\mu) = \rho$.

Proof. This is a straightforward application of trigonometric properties.

$$
\begin{aligned}
-\cos(\frac{\pi}{k}\mu) &= -\cos\left(E\left[\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q_l')\right]\right) \\
&= -\cos\left(\arcsin(\rho) + \frac{\pi}{2}\right) \\
&= -\left(\cos(\arcsin(\rho)) \cdot \cos\left(\frac{\pi}{2}\right) - \sin(\arcsin(\rho)) \cdot \sin\left(\frac{\pi}{2}\right)\right) \\
&= -(\sqrt{1-\rho^2} \cdot 0 - \rho \cdot 1) = \rho
\end{aligned}
$$

□

Lemma 3.2.4. *Var* $(S) \equiv Var(\sum_l^k I(q_l = q_l')) = k\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right)$

Proof. We again use trigonometric properties, and the fact that the indicator variables $I(q_l = q_l')$ are independent from each other.

$$
\begin{aligned}
\mathrm{Var}(\sum_{l=1}^{k} I(q_l = q_l')) &= k\left(E\left[(I(q_l = q_l'))^2\right] - (E\left[I(q_l = q_l')\right])^2\right) \\
&= k\left(\left(\frac{\arcsin(\rho)}{\pi} + \frac{1}{2}\right) - \left(\frac{\arcsin(\rho)}{\pi} + \frac{1}{2}\right)^2\right) \\
&= k\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right)
\end{aligned}
$$

□

We can then obtain a bound on the absolute bias (and thus bias) of $\hat{\rho}$, noting that $|\mathrm{Bias}(\hat{\rho})| \equiv |E[\hat{\rho}] - \rho|$. The latter inequality follows because $(S-\mu)^2$ is non-negative, and S

and $k\rho$ are both in $[0, k]$:

$$E[f(S)] = f(\mu) + E\left[\frac{f''(\tilde{S})}{2} \cdot (S-\mu)^2\right],$$

$$E[f(S)] - \rho = E\left[\frac{f''(\tilde{S})}{2} \cdot (S-\mu)^2\right],$$

$$|E[f(S)] - \rho| \leq \frac{1}{2} \max_{s\in[0,k]} |f''(s)| \cdot E\left[(S-\mu)^2\right]$$

Thus the second order term is $O\left(\frac{1}{k}\right)$:

$$\begin{aligned}
\frac{1}{2} \max_{s\in[0,k]} |f''(s)| \cdot E\left[(S-\mu)^2\right] &= \frac{1}{2} \max_{s\in[0,k]} |f''(s)| \mathrm{Var}(S) \\
&= \max_{s\in[0,k]} \left| -\frac{\pi^2}{2k^2} \cos\left(\frac{\pi}{k}s\right) \right| \cdot k \left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4} \right) \\
&= \frac{\pi^2}{2k} \cdot \rho \cdot \left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4} \right) \\
&= \frac{\rho}{2k} \left(\frac{\pi^2}{4} - \arcsin^2(\rho) \right)
\end{aligned}$$

And so putting together these pieces, we have

$$|E(\hat{\rho}_{q_M}) - \rho| \leq \frac{\rho}{2k} \left(\frac{\pi^2}{4} - \arcsin^2(\rho) \right) \text{ , or}$$

$$E(\hat{\rho}_{q_M}) = \rho + O\left(\frac{1}{k}\right)$$

□

Theorem 3.2.5. *For the 1-bit quantized MLE,* $\mathrm{Var}\left(\hat{\rho}_{q_M}\right) = \frac{1-\rho^2}{k}\left(-\arcsin^2(\rho) + \frac{\pi^2}{4}\right) = O\left(\frac{1}{k}\right)$

Proof. First, some previously-calculated values that we will use:

$$E[S] \equiv E\left[\sum_{l}^{k} I(q_l = q'_l)\right] = k \cdot \left(\frac{\arcsin(\rho)}{\pi} + \frac{1}{2}\right)$$

$$\text{Var}[S] \equiv \text{Var}(\sum_{l}^{k} I(q_l = q'_l)) = k\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right)$$

$$E[S^2] = k\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right) + k^2 \cdot \left(\frac{\arcsin(\rho)}{\pi} + \frac{1}{2}\right)^2$$

$$= k \cdot \left[(k-1)\left(\frac{\arcsin^2(\rho)}{\pi^2}\right) + k\left(\frac{\arcsin(\rho)}{\pi}\right) + \frac{k+1}{4}\right]$$

We now begin by simplifying $\text{Var}(\hat{\rho})$:

$$\text{Var}(\hat{\rho}) = Var\left(\cos\left(\pi - \frac{\pi}{k}\sum_{l}^{k} I(q_l = q'_l)\right)\right) \tag{3.9}$$

$$= \text{Var}\left(\cos\pi \cdot \cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right) - \sin\pi \cdot \sin\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right) \tag{3.10}$$

$$- 2\ \text{Cov}\left(\cos\pi \cdot \cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right) - \sin\pi \cdot \sin\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right) \tag{3.11}$$

$$= \text{Var}\left(-1 \cdot \cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right) - 0\right) - 0 \tag{3.12}$$

$$= \text{Var}\left(\cos\left(\frac{\pi}{k}\sum_{l=1}^{k} I(q_l = q'_l)\right)\right) \tag{3.13}$$

We use the Delta Method on this simplified form. Recalling that $f(S) = cos(\frac{\pi}{k}S)$, we take a second order Taylor expansion around $E(S)$, using the Lagrange remainder as described

in Appendix A.5.

$$Var\left(f(S)\right) = Var\left(f(E(S) + f'(E(S)(S - E(S)) + \frac{(S - E(S))^2}{2} f''(\tilde{S})\right)$$

$$= 0 + \text{Var}(S - E(S))(f'(E(S)))^2 + \frac{1}{4}(f''(\tilde{S}))^2 \text{Var}\left[(S - E(S))^2\right]$$

$$= \text{Var}(S)(f'(E(S)))^2 + \frac{1}{4}(f''(\tilde{S}))^2 \text{Var}\left[(S - E(S))^2\right]$$

The leading term is $\text{O}(\frac{1}{k})$:

$$\text{Var}(S)\left(f'(E(S))\right)^2 = k\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right) \cdot (-\frac{\pi}{k}\sqrt{1-\rho^2})^2$$

$$= k\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right) \cdot \frac{\pi^2}{k^2}(1 - \rho^2)$$

$$= \frac{(1-\rho^2)}{k}\left(-\arcsin^2(\rho) + \frac{\pi^2}{4}\right)$$

Noting that $\tilde{S} \in [E(S), S]$ and $f''(\tilde{S}) = O\left(1\right)$, the remainder term can be broken down into the following inequality:

$$\frac{1}{4} f''(E(\tilde{S}))^2 \text{Var}\left[(S - E(S))^2\right]$$

$$\leq \frac{1}{4} f''(\tilde{S})^2 \left(E[(S - E(S))^4] + \text{Cov}\Big((S - E(S), (S - E(S))^2\Big)\right)$$

We now apply a sub-Gaussian argument to demonstrate that this remainder term is below $\text{O}(\frac{1}{k})$ [41]. First, note that S is the sum of i.i.d sub-Gaussian random variables, and is thus sub-Gaussian itself:

$$S = \sum_{l=1}^{k} I(q_l = q_l')$$

Then it follows that $E[(S - E(S)] = O\left(\frac{1}{\sqrt{k}}\right)$, and so $E[(S - E(S)]^4 = O\left(\frac{1}{k^2}\right)$.

For the covariance term, we rely on the Cauchy-Schwartz inequality, which states that for any random variables X and Y with finite second moment,

$$E|XY| \leq \sqrt{E(X^2)}\sqrt{E(Y^2)}$$

So we have

$$\begin{aligned}
&\mathrm{Cov}\Big((S - E(S), (S - E(S))^2\Big) \\
&\quad = E\left[(S - E(S)\left((S - E(S))^2 - E([(S - E(S))^2])\right)\right] \\
&\quad \leq \sqrt{E[(S - E(S))^2]}\sqrt{[(S - E(S))^4]} \\
&\quad = O\left(\frac{1}{\sqrt{k}}\right) \cdot O\left(\frac{1}{k}\right) = O\left(\frac{1}{k^{3/2}}\right)
\end{aligned}$$

Thus our remainder terms are of order $O\left(\frac{1}{k^{3/2}}\right)$ and $O\left(\frac{1}{k^2}\right)$. Plugging this all in to (3.9), we have:

$$\mathrm{Var}(\hat{\rho}) = \frac{\pi^2(1-\rho^2)}{k}\left(-\frac{\arcsin^2(\rho)}{\pi^2} + \frac{1}{4}\right) + O\left(\frac{1}{k^{3/2}}\right) + O\left(\frac{1}{k^2}\right)$$

□

Simulated results, Quantization

We support our second order estimate calculations with plots of simulated results of $\hat{\rho}_{q_M}$ for k from 1000 to 10000 alongside the theoretical results calculated above, Fig. 3.1. Note that simulation variance is high at $O\left(\frac{1}{n_{sim}}\right)$ (n_{sim} being the number of Monte Carlo simulations), which is significant when the bias itself is $O\left(\frac{1}{k}\right)$. It is nonetheless clear that both bias and variance are of order $O\left(\frac{1}{k}\right)$, and thus so is $MSE(\hat{\rho})$.

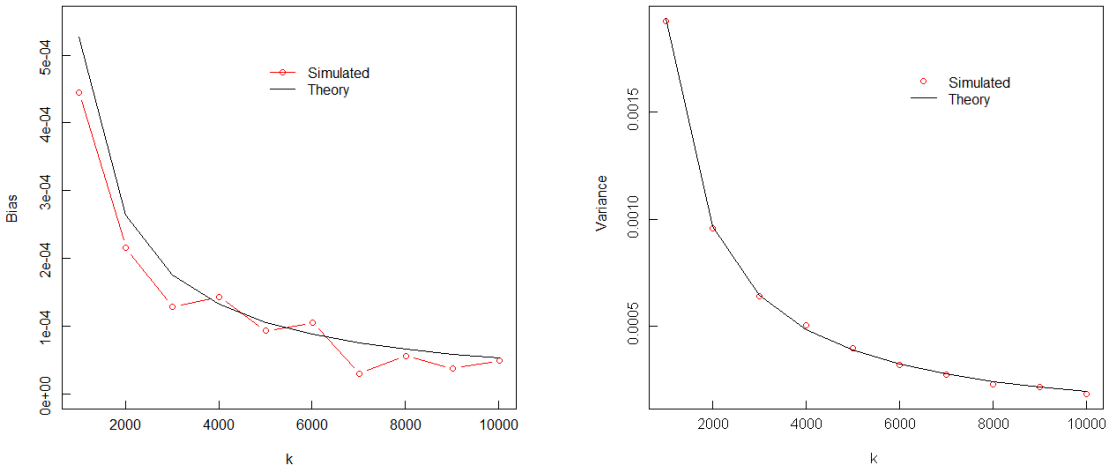


Figure 3.1: Simulated $\hat{\rho}_{q_M}$ results using 1-bit MLE demonstrating that both are $O\left(\frac{1}{k}\right)$.

3.2.3 $\hat{\rho}_{q_S}$, Stochastic Quantization estimator

We now move on to estimation of ρ after Stochastic Quantization estimation. Part of the proof below relies on Isserlis' Theorem (full proof in Appendix A.2). We simply state the results used here, where Z, Z' are each univariate Gaussian random variables with 0 mean, 1 variance, and ρ correlation:

$$E(Z^2 Z'^2) = \text{Var}(Z)\text{Var}(Z') + 2\text{Cov}(Z, Z')^2 = 1 + 2\rho^2 \qquad E(Z^2 Z') = 0$$

Analysis of $\hat{\rho}_{q_S}$, 2-level estimator

We begin with analysis of the simplest 2-level (1-bit) case. To simplify the notation for this case, we label the two bin thresholds $B_1 \equiv a$ and $B_2 \equiv b$.

For this theoretical work, we assume $b \geq \max_{i\in[1,n],j\in[1,k]}|z_{ij}|$. This can be done by setting b to the maximum of the data set we are quantizing. In practice this may not be easy to do over an entire data set of n observations, even with a tractable post-projection dimension k, as this operation is of order $n \cdot k$. It is possible to control the bias by fixing b sufficiently large so that the probability of any given observation is negligible. For example, we can set b proportional to $\sqrt{\log(nk)}$, which is of order $E[\max_{1\leq i\leq n}|Z_i|]$ so that $P(|(z_{(1)})|> b) < \alpha$ for some sufficiently low α, such as $\alpha < 1/nk$. We discuss this in more detail in Appendix A.5.

Theorem 3.2.6. *Bias*$(\hat{\rho}) = 0$

Proof. We use conditioning: $E(QQ') = E[E(QQ'|Z, Z')]$. Note that this proof applies to any number of bits.

$$E\left[QQ'|Z, Z'\right] = b^2 P(Q = b, Q' = b) + ab \cdot P(Q = a, Q' = b) + ba \cdot P(Q = b, Q' = a)$$

$$+ a^2 P(Q = a, Q' = a)$$

Now, since we set $a = -b$:

$$E\left[QQ'|Z, Z'\right] = \frac{1}{4b^2}[b^2(Z+b)(Z'+b) + b^2(b-Z)(b-Z') - b^2(Z+b)(b-Z') + b^2(Z'+b)(b-Z)]$$

$$= Z'(b + Z - b + Z)/2 = ZZ'$$

This gives us:

$$E\left[E\left[QQ'|Z, Z'\right]\right] = E(ZZ')$$

$$= Cov(ZZ') - E(Z)E(Z') = \rho$$

□

Theorem 3.2.7. *Var* $(QQ') = b^4 - \rho^2$

Proof. We begin with a lemma, proof of which is straightforward and can be found in Appendix A.

Lemma 3.2.8. $Var[E\left[QQ'|Z, Z'\right]] = 1 + \rho^2$

Now, using the above lemmas, we calculate the variance of $Q_S(\rho) = QQ'$. We condition Q, Q' on Z, Z' and calculate each term separately:

$$Var(QQ') = E(Var(QQ'|Z, Z')) \tag{3.14}$$

$$+ Var(E(QQ'|Z, Z')) \tag{3.15}$$

We first calculate term (3.14) by expanding the variance:

$$(3.14) = E\left[Var(QQ'|Z, Z')\right] \tag{3.16}$$

$$= E\Big[E[(Q^2Q'^2|Z, Z')] \tag{3.17}$$

$$- (E\left[QQ'|Z, Z'\right])^2\Big] \tag{3.18}$$

Using conditional independence of $Q, Q'|Z, Z'$, we can calculate the subsidiary terms of (3.14):

$$(3.17) = E\Big[E[(Q^2Q'^2|Z, Z')]\Big] = E\Big[E[Q^2|Z]\Big] \cdot E\Big[E[Q'^2|Z]\Big]$$

$$= b^2 \cdot b^2 = b^4$$

$$(3.18) = E\left[(E\left[QQ'|Z, Z'\right])^2\right] = E\left[(ZZ')^2\right]$$

$$= 1 + 2\rho^2 \text{ using Isserlis' Theorem as noted above}$$

Thus, in all (3.14) becomes

$$Var(QQ') = E(Var(QQ'|Z, Z')) + Var(E(QQ'|Z, Z'))$$

$$= b^4 - (1 + 2\rho^2) + (1 + \rho^2) = b^4 - \rho^2$$

Recall that the estimator is $\hat{\rho}_{q_S} = \frac{1}{k}\sum_{l=1}^{k} q_l \cdot q'_l$, where the $(q_l, q'_l) \sim (Q, Q')$ and are i.i.d., and so it follows that

$$\text{Var}(\hat{\rho}_{q_S}) = \frac{Var(QQ')}{k} = \frac{b^4 - \rho^2}{k}$$

□

3.2.4 Analysis of $\hat{\rho_S}$, estimator after Stochastic 4-level / 2-bit Quantization

We now extend the above to 2 bits. As before, since the levels are centered on 0, $\rho_{\hat{S2Q}}$ is unbiased for any number of bits. We proceed to examine the variance of $\hat{\rho}$

Theorem 3.2.9. *For the 4-level estimator, we have the following expression for Var (QQ'):*

$$\begin{aligned} Var(QQ') = a^4 P(Z, Z \in (-a, a]) + 4a^2 (E[Z \cdot I_{Z \in (-a,a), Z' > a}(a+b)] \\ - ab \cdot P(Z \in (-a, a), Z' > a)) + 2E[ZZ' I_{Z>a, Z'>a}(a+b)^2 \\ - (Z + Z')(a+b)ab \cdot I_{Z>a, Z'>a}] + 2a^2 b^2 \cdot P(Z > a, Z' > a) \\ - 2E[ZZ' I_{Z>a, Z'<-a}(a+b)^2 + (a+b)ab(Z - Z') \cdot I_{Z>a, Z'<-a} \\ - a^2 b^2 I_{Z>a, Z'<-a}] - \rho^2 \end{aligned}$$

Proof.

$$Var(QQ') = E(Var(QQ'|Z, Z')) \tag{3.19}$$

$$+ Var(E(QQ'|Z, Z')) \tag{3.20}$$

We first calculate term (3.19) by expanding the variance:

$$(3.19) = E\left[Var(QQ'|Z, Z')\right] \tag{3.21}$$

$$= E\Big[E[(Q^2 Q'^2|Z, Z')] \tag{3.22}$$

$$- (E\left[QQ'|Z, Z'\right])^2\Big] \tag{3.23}$$

We condition as in the previous section, and we will build up in the same way. Due to the conditional independence of $Q|Z, Z'$ and $Q'|Z, Z'$, we have the following equalities

(once again assuming that b is set to the maximum of the projected data).

$$(3.20) \equiv Var(E[QQ'|Z, Z']) = Var(E[Q|Z]E[Q'|Z']) = 1 + \rho^2$$

$$(3.23) \equiv E\Big[E(QQ'|Z, Z')^2\Big] = E\Big[Z^2 Z'^2\Big] = 1 + 2\rho^2$$

Now, let $f(a)$ be the p.d.f. of $Z \sim N(0, 1)$ at a, and let $\Phi(a)$ be the c.d.f. of Z at a. We again assume $\Phi(b) = 1$ and $f(b) = 0$, then using the formula for truncated normal probabilities [42]:

$$E[X|X > a] = \mu + \frac{\sigma f(a)}{1 - \Phi(a)}$$

Then we can calculate $E[E(Q^2|Z)]$, which we will later plug in to (3.22).

$$\begin{aligned} E[E(Q^2|Z)] &= 2b^2(I(Z > b)) + a^2 I(Z \in (-a, a]) + (a + b)\frac{f(a) - f(b)}{\Phi(b) - \Phi(a)}(\Phi(b) - \Phi(a)) \\ &\quad - ab(\Phi(b) - \Phi(a)) - (\Phi(b) - \Phi(a))ab \\ &\quad - (a + b)\frac{f(-b) - f(-a)}{\Phi(-a) - \Phi(-b)}(\Phi(-a) - \Phi(-b)) \\ &= 2b^2(1 - \Phi(b)) + a^2(2\Phi(a) - 1) - 2ab(1 - \Phi(a)) + 2(a + b)f(a) \end{aligned}$$

We then substitute this in to (3.22) to obtain $E(E(Q^2Q'^2|Z, Z'))$. We present the final

result of (3.22) here as the full calculation is rather long; it can be found in Appendix A.7.

$$
\begin{aligned}
(3.22) &= E(E(Q^2Q'^2|Z,Z')) \\
&= a^4P(Z,Z\in(-a,a]) + 4a^2(E[Z\cdot I_{Z\in(-a,a),Z'>a}(a+b)] \\
&\quad - ab\cdot P(Z\in(-a,a),Z'>a)) + 2E[ZZ'I_{Z>a,Z'>a}(a+b)^2 \\
&\quad - (Z+Z')(a+b)ab\cdot I_{Z>a,Z'>a}] + 2a^2b^2\cdot P(Z>a,Z'>a) \\
&\quad - 2E[ZZ'I_{Z>a,Z'<-a}(a+b)^2 + (a+b)ab(Z-Z')\cdot I_{Z>a,Z'<-a} - a^2b^2I_{Z>a,Z'<-a}]
\end{aligned}
$$

In all, we have

$$\mathrm{Var}(QQ') = E(E(Q^2Q'^2|Z,Z')) - \rho^2 \tag{3.24}$$

which concludes our proof. Again recall that the actual estimator for ρ is $\hat{\rho}_{q_S} = \frac{1}{k}\sum_{l=1}^{k} q_l\cdot q'_l$, and so $\mathrm{Var}(\hat{\rho}_{q_S}) = Var(QQ')/k$. □

As this expression is not closed form, to help confirm that our calculations are correct we simulated $\hat{\rho_S}$ with 4 levels and plotted its variance against the theoretical value we calculated above. In the next section we extend this to the variance for the general m-level estimator.

3.2.5 General m-level estimator $\hat{\rho_S}$

Theorem 3.2.10. *For general m-level (or 2^b bit) estimator $\hat{\rho_S}$,*

$$Var(QQ') = \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2B_j^2\cdot\frac{|Z-B_i|}{d}\cdot\frac{|Z'-B_j|}{d}\cdot P(Z\in(B_{i-1},B_{i+1}),Z'\in(B_{j-1},B_{j+1})) - \rho^2$$

Proof. Bias and the first three terms of variance are the same as above, so we skip to

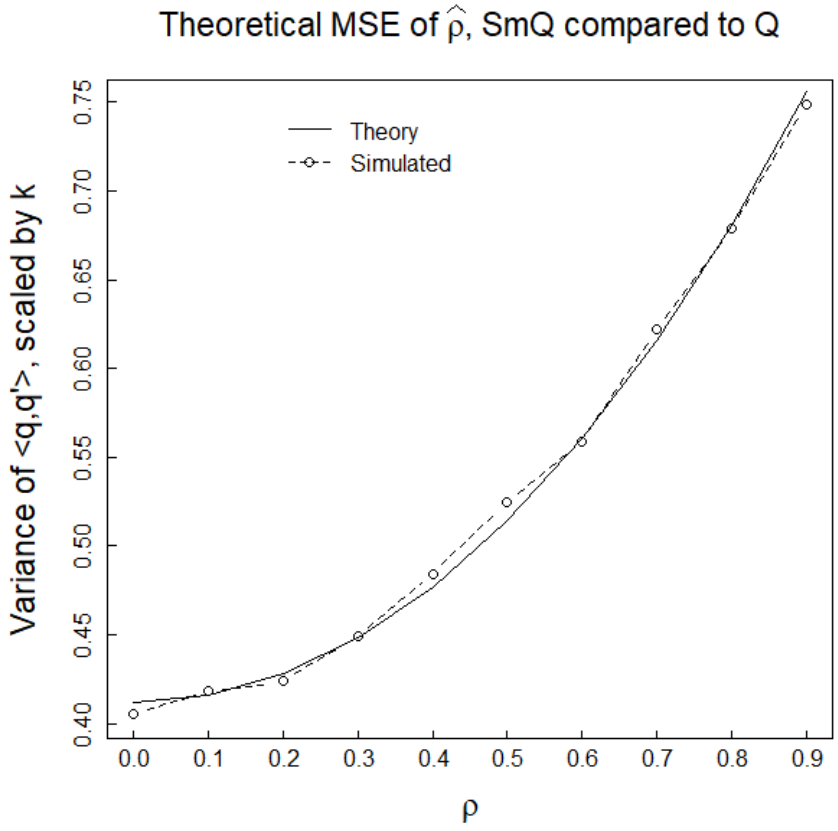


Figure 3.2: Variance of 2-bit $\hat{\rho}_S$, levels = $\{-4.0, -0.5, 0.5, 4.0\}$ corresponding to a = 0.5, b = 4.0. Theoretical and simulated values are plotted against each other, with the simulated values confirming that the variance matches (3.24).

$\mathrm{E}[Q^2Q'^2]$, which is part of our bound for general m-level stochastic quantization. Let $B_1, ..., B_m$ be the m values Q and Q' can take on. For convenience in notation and calculation we assume equidistant cut points with distance d, but this need not be the case.

$$E[Q^2Q'^2] = \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot P(Q = B_i, Q' = B_j) \tag{3.25}$$

This requires calculation of $P(Q = B_i, Q' = B_j)$:

$$P(Q = B_i, Q' = B_j) = P(Q = B_i, Q' = B_j, Z\ in(B_{i-1}, B_{i+1}, Z'\ in(B_{j-1}, B_{j+1}) \tag{3.26}$$

$$= P(Q = B_i, Q' = B_j | Z\ in(B_{i-1}, B_{i+1}, Z'\ in(B_{j-1}, B_{j+1}) \tag{3.27}$$

$$\cdot P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1})) \tag{3.28}$$

$$= \frac{|Z - B_i|}{d} \cdot \frac{|Z' - B_j|}{d} \tag{3.29}$$

$$\cdot P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1})) \tag{3.30}$$

Then equation (3.25) is:

$$E[Q^2Q'^2] = \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot \frac{|Z - B_i|}{d} \cdot \frac{|Z' - B_j|}{d} \cdot \tag{3.31}$$

$$P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1})) \tag{3.32}$$

Where this last term is easily calculated via numerical integration. Putting all of these pieces

together we get a lengthy but calculable variance for general m-level stochastic quantization:

$$Var(\hat{\rho_S}) = \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot \frac{|Z - B_i|}{d} \cdot \frac{|Z' - B_j|}{d} \cdot P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1}))$$

$$- (1 + 2\rho^2) + (1 + \rho^2)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot \frac{|Z - B_i|}{d} \cdot \frac{|Z' - B_j|}{d} \cdot P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1}))$$

$$- \rho^2$$

□

3.2.6 MSE

We can now compare the mean squared error (MSE), $MSE(\hat{\rho}) = Bias^2(\hat{\rho}) + Var(\hat{\rho})$, of our estimators. As we can see from the plot above, the MSE of Quantization is lower than that of SQ for $\rho > 0.7$ even using only 1 bit vs 3 bits (recalling that to get m levels requires $b = \log_2 m$ bits).

$$MSE(\hat{\rho}_{q_S}) = Bias^2(\hat{\rho}_{q_S}) + Var(\hat{\rho}_{q_S})$$

$$= 0 + \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot \frac{|Z - B_i|}{d} \cdot \frac{|Z' - B_j|}{d}$$

$$P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1})) - \rho^2$$

For the Quantized MLE (1-bit):

$$\text{Bias} = \frac{\rho}{2k}\left(\arcsin^2(\rho) - \frac{\pi^2}{4}\right)$$

$$\text{Var}[\hat{\rho}_{q_M}] = \frac{1}{k}(1-\rho^2)\left(\frac{\pi^2}{4} - \arcsin^2(\rho)\right) + O\left(\frac{1}{k^2}\right)$$

$$\text{MSE} = \frac{\rho^2}{4k^2}\left(\arcsin^2(\rho) - \frac{\pi^2}{4}\right)^2 + \frac{1}{k}(1-\rho^2)\left(\frac{\pi^2}{4} - \arcsin^2(\rho)\right) + O\left(\frac{1}{k^2}\right)$$

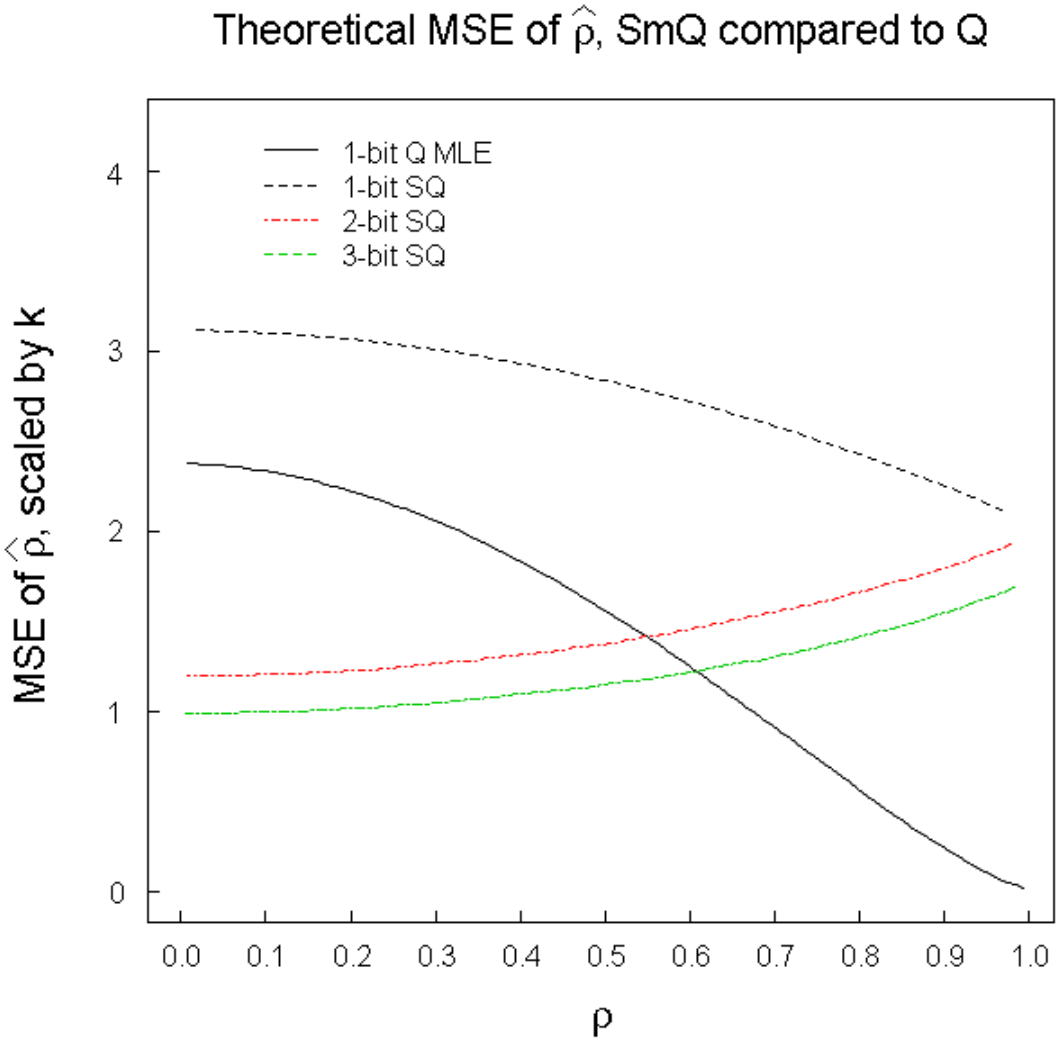


Figure 3.3: Theoretical MSE of $\hat{\rho}$ by ρ (uncompressed data's correlation), demonstrating that 1-bit Discrete Quantization has a lower MSE than 1-bit Stochastic Quantization, and still better than 2- and 3-bit Q_s for $\rho < 0.6$.

3.3 Other estimators of ρ in the presence of quantization

When estimating pairwise distances and inner products after quantized random projection, we estimate based on the quantized projected data (i.e., we do not quantize the estimators). Due to the non-linearity of the quantization function, any estimator using quantized data will be biased. The estimators we will work with can be classified into two categories: those that use the codes μ_i and those that use the bins $[t_i, t_{i+1})$. Within the first category, using codes, there are several that can be described as closed form, a cubic estimator [7], and of course the stochastic quantization estimator $\hat{\rho}$ explored in the section above. Within the second category, using bins, are the Maximum Likelihood Estimator (MLE) (also discussed in detail in section 3.2) and estimators using the Hamming distance, which we address briefly later.

Quantized estimators using codes

We begin with estimators that use codes, which correspond directly to the non-quantized estimators above: the estimators can be modified by simply using q (the coded values of z) in place of z above. We use the same names here for simplicity, and assume that when quantized data is present the quantized versions of the estimators are used.

$$(i)\ \hat{\rho}_{qlin} = \langle q_i, q_j \rangle$$

$$(ii)\ \hat{\rho}_{qadd} = c - \frac{1}{2}\|q_i - q_j\|^2$$

$$(iii)\ \hat{\rho}_{qmult} = \frac{\langle q_i, q_j \rangle}{\|q_i\|\|q_j\|}$$

where c is a constant that merely affects the bias of $\hat{\rho}_{add}$, and can simply be set so that $\mathrm{E}(\hat{\rho}_{qadd}) = \mathrm{E}(\hat{\rho}_{qlin})$ [43]. Similarly to the non-quantized case, the above estimators have straightforward and easy-to-calculate closed forms.

In [7], the authors note that, under Lloyd-Max quantization using b bits, the bias of

the linear estimator $\hat{\rho}_{qlin}$ is bounded by $2\pi\sqrt{3}\frac{\rho^2}{2^{2b}}$. Crucially, this means that the bias has exponential rate of decay, i.e. $O(2^{-2b})$. The other estimators improve bias and / or variance under certain settings, primarily depending on ρ. The paper goes on to explore the trade off between storage bits b and reduced dimension k in greater detail, assuming that storage is limited to some total bits $B = kb$. Part of our research will involve exploring this relationship in applications.

We can also substitute the codes q in place of z into the MLE described above. Because this estimator is no longer the MLE, we call it $\hat{\rho}_{cubic}$ instead. As before, $\hat{\rho}_{cubic}$ is

$$\hat{\rho}_{cubic} = \underset{r}{argmin}\left(-\frac{1}{2}\log(1-r^2) - \frac{1}{2}\frac{1}{1-r^2}(\|q_i\|^2 + \|q_j\|^2 - 2r\langle q_i, q_j\rangle)\right)$$

Quantized estimators using bins

We now briefly discuss estimators that do not require an encoding of the quantization, but instead rely on collisions within the bins: that is, whether or not the quantized values are in the same bin. These estimators are the Maximum Likelihood Estimator in the quantized context (as opposed to the MLE for the full-data context), and a Hamming distance-based estimator.

We begin by describing the MLE. Without going into too much detail at this stage, we observe that there are $2 \cdot 2^b$ bin pairings possible given a pair of values using b-bit quantization. The MLE counts these distinct empirical cell frequencies and utilizes symmetries to reduce the number of cells to L, where $L = (2^{b-1})(2^{b-1}+1)$. Calling each cell probability π_l and the empirical analogues $\hat{\pi}_l$, we have the negative log-likelihood $\mathcal{L}(\rho) = -\sum_{l=1}^{L}\hat{\pi}_l \log(\pi_l(\rho))$ [19], [44]. Then

$$\hat{\rho}_{q_M} = \underset{\rho}{argmin}(-\sum_{l=1}^{L}\hat{\pi}_l \log(\pi_l(\rho)))$$

A simpler method uses the Hamming distance, which counts the number of differing elements in the vectors:

$$d_h(q_i, q_j) = \frac{1}{k} \sum_{l=1}^{k} I(q_{il} \neq q_{jl})$$

Then let $\theta(\rho)$ be the collision probability, i.e., a function such that

$$\rho \mapsto E_\rho(I_{Q(Z)=Q(Z')}) = P_\rho(Q(Z) = Q(Z'))$$

It follows that $1 - d_h(q_i, q_j)$ is an estimator of $\theta(\rho)$, i.e. we can call

$$\hat{\theta}(\rho) \equiv 1 - d_h(q_i, q_j)$$

Thus we can construct an estimator

$$\hat{\rho}_{ham} = \theta^{-1}(\hat{\theta}) = \theta^{-1}(1 - d_h(q_i, q_j))$$

Note that the Hamming distance is one-to-one and monotone decreasing on $\rho \in [0, 1]$, while $\theta(\rho)$ monotone increasing on $\rho \in [0, 1]$, thus both are invertible [45].

3.4 Application to Spectral Clustering

We now move on from the previous sections and discuss an application of quantized random projections to spectral clustering. Spectral clustering is a clustering algorithm that uses the eigenvalues of the data's graph similarity matrix. It is a relaxation of the graph cut problem [1]. One of its advantages is that it relies only on pairwise distances; since this is what random projection preserves, we would expect spectral clustering to be a good application of RP. We conducted a base experiment of just spectral clustering and random projection, which can be found in the Appendix B, and contains a more detailed description of the experiment. In this section we combine spectral clustering with each of two methods of dimensionality reduction, random projection and principal component analysis, as well as each of our two methods of data compression (quantization or Stochastic Quantization).

The spectral clustering algorithm we used was introduced in [46]. We followed the implementation outlined in [1] with minimal modification to incorporate our dimensionality reduction. Our goal is to cluster data X to κ clusters.

3.4.1 Quantized experiments

We conduct an experiment using dimensionality reduction combined with spectral clustering, where we apply both random projection and PCA to the Modified National Institute of Standards and Technology (MNIST) database [47]) of handwritten digits, each represented by a 28×28 grey scale bitmap. Essentially our methodology is as follows: we project data $X \in \mathbb{R}^{n \times d}$ using RP (or PCA for comparison) to obtain $Z = XR, z_i \in \mathbb{R}^k$. For each compression type we then perform spectral clustering. For details on the steps therein, as well as parameter settings and findings, please see Appendix B.

We assess the clustering performance using several measures, including Rand Index [48] to measure clustering similarity (higher indicates greater similarity and thus better performance), ratio cut (lower values of ratio cut indicate more dissimilar partitions, i.e. better performance [1]), as well as straightforward classification accuracy when it is obtainable.

For this stage of our experiment, we quantize the projected data Z in Euclidean space $\mathbb{R}^k$ to quantized alphabet $(\mathcal{M}^{\pm})^k$ using deterministic quantization.

We ran this experiment on the MNIST data set. Please see Appendix B for a full description of the experiment, with emphasized changes to the algorithm in bold. In Fig 3.4, we can see that classification accuracy using quantized data converges quickly to full data results.

3.5 Discussion

In this chapter we established bounds on estimation of ρ after two quantization methods. While stochastic quantization is unbiased (so long as bins are defined post-compression, which may not be feasible), the variance of deterministic quantization is significantly lower than that of stochastic quantization. In all, deterministic quantization's MSE is much lower than that of stochastic quantization for most levels of ρ and b bits. In the following chapter, we expand this work to exploring the relationship between random projection and spectral clustering. We eventually bring in quantization as well, though going forward the focus is only on deterministic quantization.

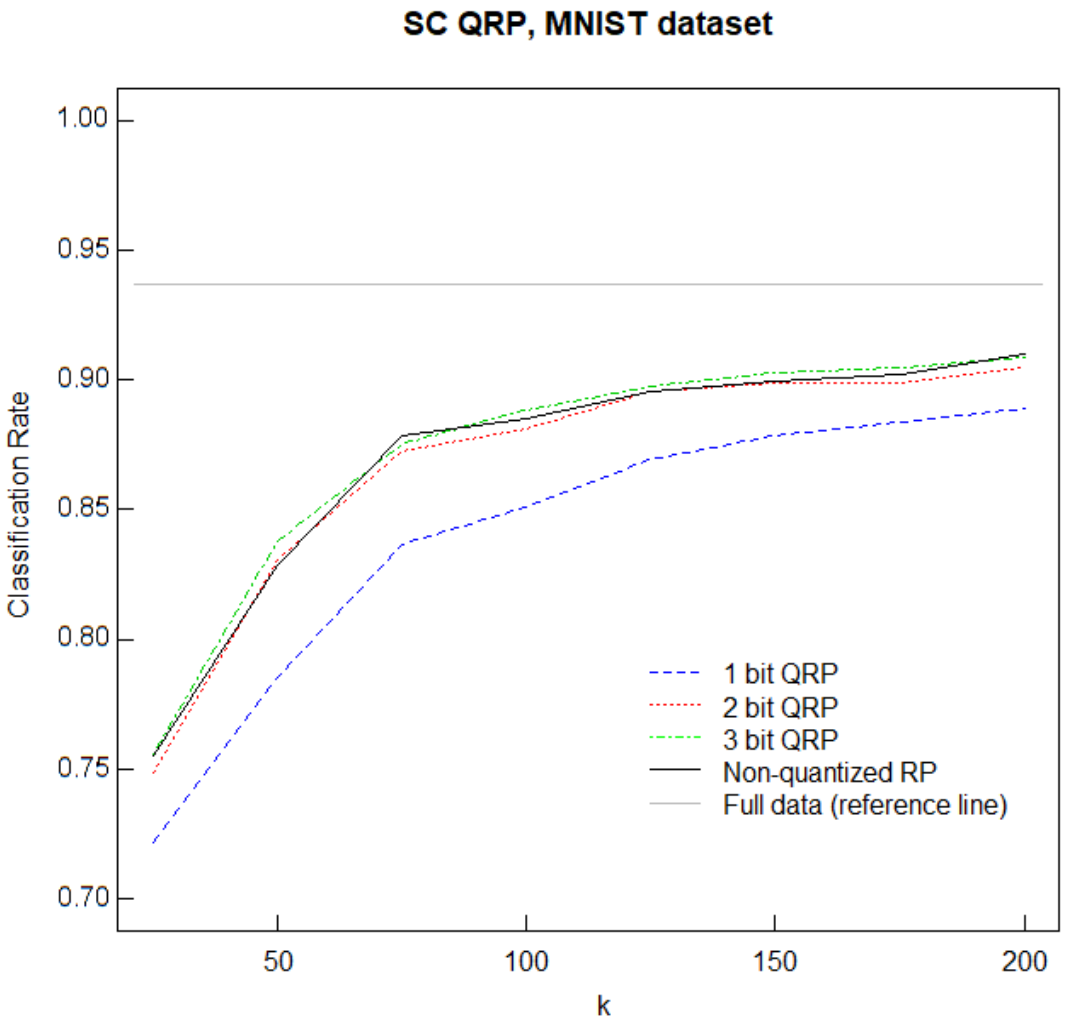


Figure 3.4: Classification Accuracy of QRP + SC on MNIST, averaged results over all pairs. We can see that quantized performance rapidly converges to that of full precision with only 2 or 3 bits.

Chapter 4: Theory of Random Projection with Spectral Clustering

We now move on to theoretical bounds of random projection. This chapter establishes bounds on the mean squared error, $\text{MSE}(\hat{\rho}) = \text{Bias}(\hat{\rho})^2 + \text{Var}(\hat{\rho})$, of our estimators $\hat{\rho}_{lin} = \langle z, z' \rangle$ and $\hat{\rho}_{mult} = \frac{\langle z, z' \rangle}{\|z\|\|z'\|}$. We spend most of our time in the full data context before adding quantization to the mix. We establish that, for both estimators, $\text{Bias}(\hat{\rho}) = O\left(\frac{1}{k}\right)$, and thus $\{\text{Bias}(\hat{\rho})\}^2 = O\left(\frac{1}{k^2}\right)$, and also establish that $\text{Var}(\hat{\rho}) = O\left(\frac{1}{k}\right)$. Note that we assume unit vectors, i.e. $\|x\|^2 = 1$ for all observations x.

Chapter outline

This chapter has been divided into two sections, one for each of $\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$. Within each section we begin with bounds for $\hat{\rho}$ before exploring how that interacts with Gaussian similarity measure $g(\rho) = \exp\left(\frac{\rho - 1}{\sigma^2}\right)$.

Foundational quantities, including some previous work

Recall that each element of the random projection matrix is a univariate Gaussian random variable with $\frac{1}{k}$ variance, or $r_{ij} \sim N(0, \frac{1}{k})$. As before, lower case z, z' indicate an arbitrary pair of $k \times 1$ compressed observations from randomly projected, non-quantized data Z. We also use z_i to denote the i^{th} element of z, and z'_i to denote the i^{th} element of z'. Recall that $z_i \sim Z$, where $Z \sim N(0, 1)$ (given our unit vector assumption). Going forward in this chapter we use (Z, Z') to refer to arbitrary bivariate normal random variables with 0 mean,

1 variance, and correlation ρ. Then we can have several quantities we will use further on:

$$\begin{pmatrix} z_i \\ z_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

$$E[(\hat{\rho})] = E[\langle z, z' \rangle] = \langle x, x' \rangle = \rho,$$

$$Var[(\hat{\rho})] = \frac{\|x\|^2 \cdot \|x'\|^2 + \langle x, x' \rangle^2}{k} = \frac{1+\rho^2}{k}$$

$$E(z_i z_i') = \mathrm{Cov}(z_i z_i') + E(z_i)E(z_i') = \mathrm{Cov}(z_i z_i') = \rho$$

$$(i)\ g(\rho) = \exp\left(\frac{\rho - 1}{\sigma^2}\right) \qquad (ii)\ g^{(j)}(\rho)\ = \frac{1}{\sigma^{2j}} \exp\left(\frac{\rho - 1}{\sigma^2}\right)$$

4.1 Bounds for Estimators of ρ after Random Projection

In this section we establish bounds on the mean squared error (MSE) of $\hat{\rho}$ for various estimators. Our big picture goal is to establish the validity of quantized random projection as a dimenstionality reduction method: in particular, that it retains pairwise distances between any (all) observations. This demonstrates the consistency of these estimators and viability of random projection as a means of data compression. The basic flow of our work is as follows: We establish bounds on estimation of pairwise correlation ρ. We then show that these bounds are retained after the Gaussian similarity function used in spectral clustering $g(\rho) \equiv \exp((\rho - 1)/\sigma^2)$. We continue by demonstrating that the pairwise distances are still retained after deterministic quantization, both before and after applying Gaussian distance measure.

4.1.1 Variance of $g(\hat{\rho}_{lin})$

Theorem 4.1.1. *$Var(g(\hat{\rho}_{lin})) = \frac{(1+\rho^2)}{k} \cdot \frac{e^{(\rho-1)/\sigma^2}}{\sigma^4} + O\left(\frac{1}{k^2}\right)$*

Delta method / Taylor expansion for variance

We create an upper bound on $\text{Var}(g(\hat{\rho}))$ via Taylor expansion. We use the Lagrange form of the remainder after the first term, $R = \frac{1}{2}f''(E(\tilde{\rho}))(\hat{\rho} - E(\hat{\rho}))^2$.

$$\begin{aligned}\text{Var}(g(\hat{\rho})) &= \text{Var}\left[g(E(\hat{\rho})) + g'(E(\hat{\rho}))(\hat{\rho} - E(\hat{\rho}) + \frac{1}{2}g''(\tilde{\rho})(\hat{\rho} - E(\hat{\rho}))^2\right] \\ &= 0 + \text{Var}(\hat{\rho})(g'(E(\hat{\rho}))^2 + R \\ &= \frac{(1+\rho^2)}{k} \cdot \frac{e^{(\rho-1)/\sigma^2}}{\sigma^4} + R\end{aligned}$$

Where we replaced $g''(\tilde{\rho})(\hat{\rho} - E(\hat{\rho}))^2$ with the remainder R:

$$R = \left(\frac{g''(E(\tilde{\rho}))^2}{4}\right)\text{Var}(\hat{\rho} - E(\hat{\rho}))^2 + \text{Cov}\Big((\hat{\rho} - E(\hat{\rho}), (\hat{\rho} - E(\hat{\rho}))^2\Big)g'(E(\hat{\rho}))g''(\tilde{\rho}),$$

$$\tilde{\rho} = \max(\langle z, z' \rangle, \langle x, x' \rangle)$$

It is clear that the first order term is $O\left(\frac{1}{k}\right)$ with a constant in terms of ρ and σ. We now look at remainder terms individually.

Lemma 4.1.2. *The covariance term of remainder* $E(\hat{\rho} - \rho)^3 = O\left(\frac{1}{k^2}\right)$

Proof. We first expand $(\hat{\rho} - \rho)^3$ and rearrange terms. Recall that we have the following constants: $E(\hat{\rho}) = \rho$, $Var(\hat{\rho}) = \frac{1+\rho^2}{k}$.

$$E(\hat{\rho} - \rho)^3 = E(\hat{\rho}^3) - E(3\hat{\rho}^2\rho) + E(3\hat{\rho}\rho^2) - E(\rho^3) \tag{4.1}$$

Now we take a Taylor expansion of $E(\hat{\rho}^3)$, letting $f(\rho) = \rho^3$:

$$E(\hat{\rho}^3) = E\left[f(\rho) + f'(\rho)(\hat{\rho} - \rho) + \frac{f''(\tilde{\rho})(\hat{\rho} - \rho)^2}{2}\right]$$

$$= \rho^3 + 3\rho^2 E(\hat{\rho} - \rho) + \frac{6E[\tilde{\rho}]}{2} \cdot E[(\hat{\rho} - \rho)^2]$$

$$= \rho^3 + 0 + 3E[\tilde{\rho}] \cdot \frac{1 + \rho^2}{k}$$

Similarly,

$$E[\hat{\rho}^2] = (Var(\hat{\rho}) + (E[\hat{\rho}])^2) = \frac{1 + \rho^2}{k} + \rho^2$$

We plug the above results for $E[\hat{\rho}^2]$ and $E[\hat{\rho}^3]$ into 4.1:

$$E(\hat{\rho} - \rho)^3 = \rho^3 + 3E\left[\tilde{\rho}\right] \cdot \frac{1 + \rho^2}{k} - 3\rho \cdot \left(\frac{(1 + \rho^2)}{k} + \rho^2\right) + 3\rho^3 - \rho^3$$

$$= \frac{3(1 + \rho^2) \cdot E[\tilde{\rho} - \rho]}{k}$$

Now note that $\tilde{\rho} \in (\rho, \hat{\rho})$, and thus $|\tilde{\rho} - \rho\| \leq |(\hat{\rho} - \rho)|$. And so:

$$E(\tilde{\rho} - \rho) \leq E[max(\langle z, z' \rangle - \langle x, x' \rangle), 0)]$$

$$= P(\langle z, z' \rangle - \langle x, x' \rangle > 0) E(\hat{\rho} - \rho)$$

$$\leq \text{Bias}(\hat{\rho}) = O\left(\frac{1}{k}\right),$$

$$\therefore E(\hat{\rho} - \rho)^3 \leq 3(1 + \rho^2) \cdot O\left(\frac{1}{k}\right) \cdot \frac{1}{k} = O\left(\frac{1}{k^2}\right)$$

□

Simulation Results: Plot of $E(\hat{\rho}-\rho)^3$ against k

Noting that the above is an upper bound, we plot results of regression lines on simulated values of $\hat{\rho}$: We plot $E(\hat{\rho}-\rho)^3$ against k, along with regression lines against each of $1/k^{1.5}$ and $1/k^2$. This simulation and plot strongly suggests that $E(\hat{\rho}-\rho)^3$ is of order $1/k^2$.

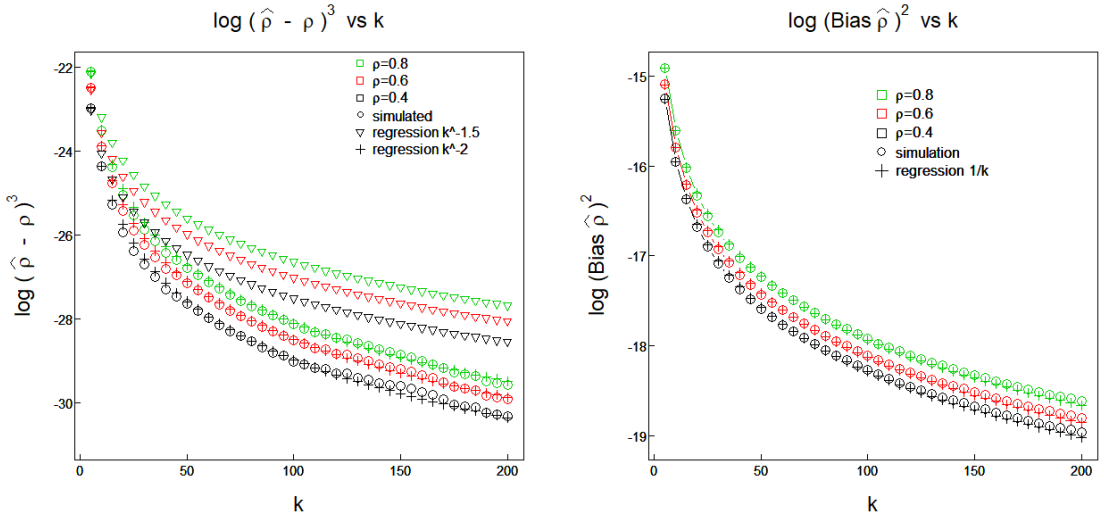


Figure 4.1: Calculated and simulated results for Taylor expansion remainder terms of Variance of $\hat{\rho}$. In these plots we plot both the actual values, along with plotted curves with regression lines, verifying that (left) $\mathrm{E}(\hat{\rho}-\rho)^3 = O\left(\frac{1}{k^2}\right)$, (right) $\mathrm{Bias}^2\ (\hat{\rho})$ is $O\left(\frac{1}{k}\right)$.

Lemma 4.1.3. *$E[\mid \hat{\rho}-\rho \mid^q] = O\left(k^{-q/2}\right)$, for $q \geq 1$*

Proof. Recalling that $\hat{\rho} = \frac{1}{k}\sum_i^k z_i z_i'$, and (z_i, z_i') are i.i.d. bivariate standard normal with correlation ρ. Then $z_i z_i' - \rho$ is a 0-mean sub-exponential random variable. A 0-mean random variable V is sub-exponential [41] if and only if there exists a constant $C > 0$ such that

$$E[exp(\lambda V) \leq exp(\lambda^2 C^2) \text{ for all } \lambda \text{ s.t. } \mid \lambda \mid \leq 1/C \tag{4.2}$$

For convenience we define $\zeta_i = (z_i z_i' - \rho)/k$ so that:

$$\sum_i^k \zeta_i \equiv \sum_i^k (z_i z_i' - \rho)/k = \hat{\rho} - \rho \tag{4.3}$$

We can see that $\sum \zeta_i$ is itself sub-exponential with constant $C/\sqrt{k}$:

$$E\left[\exp(\lambda \sum_i^k \zeta_i\right] = E\left[\exp(\lambda \sum_i^k \frac{z_i z_i' - \rho}{k}\right] = \left(\exp\left(\frac{C^2}{k^2}\lambda^2\right)\right)^k = \exp\left(\left(\frac{C\lambda}{\sqrt{k}}\right)^2\right) \tag{4.4}$$

In general, for any random variable V that is sub-exponential with constant C, we have that

$$E[|V|^q] \leq (C'q)^q \text{ for some } C' = \alpha C, \alpha > 0$$

which leads to our final result:

$$E[\mid \hat{\rho} - \rho \mid^q] \leq (qC')^q k^{-q/2} = O\left(k^{-q/2}\right), q \geq 1$$

It follows that $E(\mid \hat{\rho} - \rho \mid^4) = O(k^{-2})$ □

Putting components of variance together

From the above we can now conclude that

$$\mathrm{Var}(f(\hat{\rho})) = 0 + \mathrm{Var}(\hat{\rho})(g'(E(\hat{\rho}))^2 + \left(\frac{g''(E(\tilde{\rho}))^2}{4}\right)\mathrm{Var}(\hat{\rho} - E(\hat{\rho})^2) +$$

$$\mathrm{Cov}\Big((\hat{\rho} - E(\hat{\rho})(\hat{\rho} - E(\hat{\rho}))^2\Big) g'(E(\hat{\rho}))f''(\tilde{\rho})$$

$$= \frac{(1+\rho^2)e^{(\rho-1)/\sigma^2}}{k\sigma^4} + O(1/k^2) + O(1/k^2)$$

4.1.2 Expectation and bias of $g(\hat{\rho}_{lin})$

Delta method for expectation

We now move to a Taylor expansion the the expectation. Using similar techniques as before and described in A.5, we have

$$E[g(\hat{\rho})] = g(E[\hat{\rho}]) + E\left[\frac{g''E[\hat{\rho}]}{2} \cdot (\hat{\rho} - E[\hat{\rho}])^2\right]$$

Again recall that the linear estimator is unbiased, $E(\hat{\rho}) = \rho$, and $g(\rho) = \exp((\rho - 1)/\sigma^2)$. So we can then obtain a bound on the absolute bias (and thus bias) of $\hat{\rho}$, $|\text{Bias}(g(\hat{\rho}))| \equiv |E[g(\hat{\rho})] - g(\rho)|$. Noting that $(\hat{\rho} - E[\hat{\rho}])^2$ is non-negative, and $\hat{\rho}$ and ρ are both in $[0, 1]$ and thus so is $\tilde{\rho}$, we have

$$E[g(\hat{\rho})] = g(E[\hat{\rho}]) + E\left[\frac{g''E[\hat{\rho}]}{2} \cdot (\hat{\rho} - E[\hat{\rho}])^2\right],$$

$$E[g(\hat{\rho})] - \rho = E\left[\frac{g''E[\hat{\rho}]}{2} \cdot (\hat{\rho} - E[\hat{\rho}])^2\right],$$

$$|E[g(\hat{\rho})] - \rho| \leq \frac{1}{2}\max_{r\in[0,1]}|g''(r)| \cdot \text{Var}(\hat{\rho})$$

Thus the second order term is $O\left(\frac{1}{k}\right)$:

$$\begin{aligned}\frac{1}{2}\max_{r\in[0,1]}|g''(r)| \cdot \text{Var}(\hat{\rho}) &= \max_{r\in[0,1]}\left|\frac{1}{\sigma^4}\exp((r-1)/\sigma^2)\right| \cdot \left(\frac{1+\rho^2}{k}\right) \\ &= \frac{1}{\sigma^4}\exp(1/\sigma^2) \cdot \left(\frac{1+\rho^2}{k}\right)\end{aligned}$$

And so putting together these pieces, we have

$$|E[g(\hat{\rho})] - \rho| \leq \frac{\exp(1/\sigma^2)(1+\rho^2)}{k\sigma^4} \text{ , or}$$

$$E[g(\hat{\rho})] = \rho + O\left(\frac{1}{k}\right)$$

and so

$$\text{Bias}(g(\hat{\rho}_{lin})) \leq \frac{\exp(1/\sigma^2)(1+\rho^2)}{k\sigma^4}$$

Analysis of MSE for $\hat{\rho}_{mult}$

The following two subsections analyse $\widehat{\rho}_{mult} = \frac{\langle z, z' \rangle}{\|z\|\|z'\|}$, which we refer to as $\hat{\rho}$ for convenience in this section. (Any references to other estimators in this section will be spelled out explicitly, e.g. $\hat{\rho}_{lin}$.) We go about our analysis in several stages, beginning with listing established work and known facts. We show that the bias of $\hat{\rho} = O(\frac{1}{k})$ via Delta method (Section 4.1.3), then show that the bias of $g(\hat{\rho}) = O(\frac{1}{k})$ as well (Section 4.3.2), where $g(\rho) = \exp((\rho - 1)/\sigma^2)$ is the Gaussian similarity function we use for spectral clustering. We do the same for $\text{Var}(g(\hat{\rho}))$ in Section 4.2.2.

Known quantities for proofs concerning the estimator $\hat{\rho}_{mult}$

We define and quickly calculate some quantities for use in this section. Recall that Z, Z' indicate an arbitrary pair of scalar elements from vectors z, z'.

$$\hat{\rho} = \frac{\langle z, z' \rangle}{\|z\|\|z'\|} \qquad \text{Cov}(Z, Z') = E(ZZ') = \frac{\rho}{k}$$

$$\text{Var}(Z) = \frac{\|x\|^2}{k} = \frac{1}{k} \qquad E(ZZ') = \text{Cov}(Z, Z') + E(Z)E(Z') = \frac{\rho}{k}$$

$$\text{Var}(\|z\|^2) = \frac{2}{k} \qquad \text{Var}(\hat{\rho}) = \frac{(1 - \langle x, x' \rangle^2)^2}{k} + O\left(\frac{1}{k^2}\right) = \frac{(1-\rho^2)^2}{k} + O\left(\frac{1}{k^2}\right)$$

The next few properties follow from Isserlis' Theorem applied to a bivariate normal with 0 mean, unit variance, and ρ correlation:

$$E(Z^3Z') = 3\text{Cov}(Z,Z')\text{Var}(Z) \qquad E(Z^2Z'^2) = \text{Var}(Z)\text{Var}(Z') + 2\text{Cov}(Z,Z')^2$$

$$= 3 \cdot \frac{\rho}{k} \cdot \frac{1}{k} = \frac{3\rho}{k^2} \qquad = \frac{1}{k^2} + \frac{\rho^2}{k^2} = \frac{1+\rho^2}{k^2}$$

Now the Gaussian distance of ρ is $g(\rho) = \exp(\frac{\rho-1}{\sigma^2})$, and its derivatives are:

$$(i) \quad g(\rho) = \exp\left(\frac{\rho-1}{\sigma^2}\right) \qquad (ii) \quad g^{(j)}(\rho) = \frac{1}{\sigma^{(2j)}} \exp\left(\frac{\rho-1}{\sigma^2}\right)$$

Note that we shall assume that $\sigma = 1$ for most of our work.

4.1.3 Expectation of $\hat{\rho}_{mult}$

Theorem 4.1.4. *Bias*$(\hat{\rho}) = O\left(\frac{1}{k}\right)$

Proof. Before proving this theorem, we establish some lemmas. These lemmas will use some of the below facts, which rely on a simplified expression of $\hat{\rho}$:

$$\hat{\rho} = \frac{\langle z, z' \rangle}{\|z\|\|z'\|} = \frac{a}{\sqrt{bc}} = f(a,b,c)$$

Where, for simplicity below, we re-label some terms as a, b, and c:

$$a = \langle z, z' \rangle \qquad b = \|z\|^2 \qquad c = \|z'\|$$

$$^2 \quad E(a) = \rho \qquad E(b) = \|x\|^2 = 1 \qquad E(c) = \|x'\|^2 = 1$$

Then we can calculate the gradient of f:

$$\nabla f = \begin{pmatrix} b^{-1/2}c^{-1/2} & -\frac{1}{2}ab^{-3/2}c^{-1/2} & ab^{-1/2}c^{-3/2} \end{pmatrix}^T \tag{4.5}$$

$$\nabla^2 f = \begin{pmatrix} 0 & -\frac{1}{2}b^{-3/2}c^{-1/2} & -\frac{1}{2}b^{-1/2}c^{-3/2} \\ -\frac{1}{2}b^{-3/2}c^{-1/2} & \frac{3}{4}ab^{-5/2}c^{-1/2} & \frac{1}{4}ab^{-3/2}c^{-3/2} \\ -\frac{1}{2}b^{-1/2}c^{-3/2} & \frac{1}{4}ab^{-3/2}c^{-3/2} & \frac{3}{4}ab^{-1/2}c^{-5/2} \end{pmatrix} \tag{4.6}$$

$E(ac)$ and $E(bc)$

We now progress by calculating each expectation required, beginning with $E(ac)$ and $E(bc)$ in this subsection.

Lemma 4.1.5.

$$E(ab) = E(ac) = \frac{3\rho}{k} + \rho - \frac{\rho}{k} = \rho + \frac{2\rho}{k}, \qquad E(bc) = 1 + \frac{\rho^2}{k}$$

Proof. Using Isserlis' Theorem as mentioned in Chapter 3, we have the facts that

$$E(Z^3Z') = \frac{3\rho}{k^2}, \qquad E(Z^2Z'^2) = \frac{1+\rho^2}{k^2}, \qquad E(Z^2) = \frac{1}{k}$$

Then we plug in the above to solve for $E[ab]$ and $E[ac]$

$$E[ab] = E[ac] = kE(Z^3Z') + k(k-1)E(ZZ')E(Z^2)$$

$$= k \cdot \frac{3\rho}{k^2} + (k^2 - k) \cdot \frac{\rho}{k} \cdot \frac{1}{k}$$

$$= \frac{3\rho}{k} + \rho - \frac{\rho}{k} = \rho + \frac{2\rho}{k}$$

Similarly for $E[bc]$,

$$E(bc) = E(\|z\|^2\|z'\|^2) = E\left[\left(\sum_i^k Z_i^2\right)\left(\sum_i^k Z_i^{2'}\right)\right]$$

$$= E\left[(z_1^2 + \cdots + z_k^2)(z_1'^2 + \cdots + z_k'^2)\right]$$

$$= E\left[kE(Z_1^2 Z_1'^2) + k(k-1)E(Z_1^2)E(Z_2'^2)\right]$$

$$= k\frac{1+\rho^2}{k^2} + \frac{(k^2-k)}{k^2} = 1 + \frac{\rho^2}{k}$$

□

Delta method for $E(\hat{\rho})$

We are now prepared to prove the main theorem. We take a Taylor expansion around E[$\hat{\rho}$], using the Lagrange form of the remainder setting $\tilde{a} \in [\langle z, z' \rangle, \rho], \tilde{b} \in [\|z\|^2, \|x\|^2]$, and $\tilde{c} \in [\|z'\|^2, \|x'\|^2]$. We use properties of $\hat{\rho} = \frac{a}{\sqrt{bc}}$ in equation line (4.5). The proof is outlined in a somewhat abbreviated form here; the full version can be found in Appendix A.7.1.

$$
\begin{aligned}
E\left[\hat{\rho}\right] &= E\left[\frac{\langle z, z' \rangle}{\|z\|\|z'\|}\right] = E[\frac{a}{\sqrt{bc}}] \\
&= E\left[f\left(E\begin{pmatrix} a \\ b \\ c \end{pmatrix}\right) + \frac{1}{2}\begin{pmatrix} a-\rho & b-1 & c-1 \end{pmatrix} \nabla^2 f(\tilde{a}, \tilde{b}, \tilde{c}) \begin{pmatrix} a-\rho \\ b-1 \\ c-1 \end{pmatrix}\right] \\
&\leq \rho + \\
&E\left[\begin{pmatrix} a-\rho & b-1 & c-1 \end{pmatrix} \begin{pmatrix} 0 & -\frac{1}{2}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{1}{2}} & -\frac{1}{2}\tilde{b}^{-\frac{1}{2}}\tilde{c}^{-\frac{3}{2}} \\ -\frac{1}{2}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{1}{2}} & \frac{3}{4}\tilde{a}\tilde{b}^{-\frac{5}{2}}\tilde{c}^{-\frac{1}{2}} & \frac{1}{4}\tilde{a}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{3}{2}} \\ -\frac{1}{2}\tilde{b}^{-\frac{1}{2}}\tilde{c}^{-\frac{3}{2}} & \frac{1}{4}\tilde{a}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{3}{2}} & \frac{3}{4}\tilde{a}\tilde{b}^{-\frac{1}{2}}\tilde{c}^{-\frac{5}{2}} \end{pmatrix} \begin{pmatrix} a-\rho \\ b-1 \\ c-1 \end{pmatrix}\right] \\
&= \rho + \frac{2\rho + \tilde{a}(3 + \rho^2/2)}{k}
\end{aligned}
$$

And thus Bias($\hat{\rho}$) = $O\left(\frac{1}{k}\right)$. □

To illustrate the relationship between bias $\hat{\rho}$ and ρ, we plot simulated and theoretical values below in Fig. 4.2. We plot log absolute bias to better highlight difference.

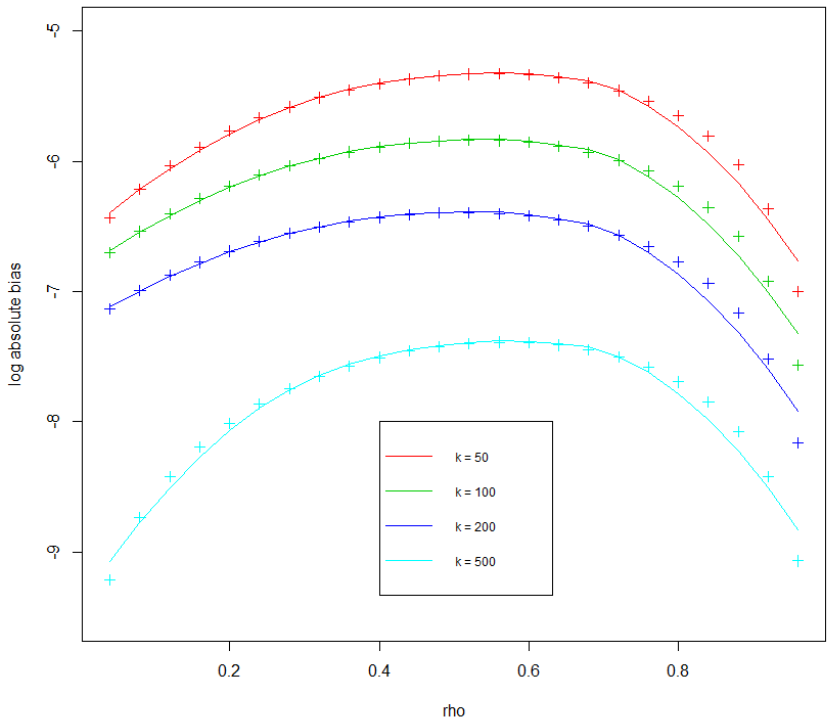


Figure 4.2: log |bias| vs ρ, separate models for each k. In this plot we demonstrate that bias $\rho = O\left(\frac{1}{k}\right)$; we plot log values to aid clarity.

4.2 Bounds for Gaussian Similarity Measure

This section again uses the Delta method / Taylor expansions, this time to calculate bounds on expectation and variance of Gaussian similarity measure $g(\hat{\rho}) = \exp((\hat{\rho}-1)/\sigma^2)$, for both the linear and multiplicative estimators discussed in section 4.1. We do this to show that the bounds we demonstrated for random projection itself are also retained for use in spectral clustering, or any other method whose primary (or even only) dependence on data is via a related distance. We go into great detail for the proofs in this section as the principles are relied upon heavily in later proofs. We begin with the multiplicative estimator $\hat{\rho}_{mult}$.

4.2.1 Expectation of $g(\hat{\rho}_{mult})$

Theorem 4.2.1. *Bias* $[g(\hat{\rho}_{mult})] = O(\frac{1}{k})$

Proof. We prove this using the delta method, using the $O\left(\frac{1}{k}\right)$ bias of $\hat{\rho}$ established above. We will specifically use these properties shown above:

$$E(\hat{\rho}) = \rho + O\left(\frac{1}{k}\right), \qquad \text{Var}(\hat{\rho}) = \frac{(1-\rho^2)^2}{k} + O\left(\frac{1}{k^2}\right)$$

$$g(\rho) = \exp((\rho-1)/\sigma^2); \qquad g(\rho) = \exp(\rho - 1) \text{ for } \sigma = 1$$

We proceed to take a Taylor expansion of $E(g(\hat{\rho}))$:

$$E(g(\hat{\rho})) = g(E(\hat{\rho})) + \frac{1}{2}g''(E(\tilde{\rho}))\text{Var}(\hat{\rho})$$

$$\text{So } E(g(\hat{\rho})) = \exp\left(\rho + O\left(\frac{1}{k}\right) - 1\right) \cdot \left(1 + \frac{1}{2}\text{Var}(\hat{\rho})\right) \text{ since } \sigma = 1$$

$$= \exp\left(\rho + O\left(\frac{1}{k}\right) - 1\right) \cdot \left(1 + \frac{(1-\rho^2)^2}{2k} + O(\frac{1}{k^2})\right)$$

$$= \exp(\rho - 1) \cdot \exp\left(O\left(\frac{1}{k}\right)\right) \cdot \left(1 + \frac{(1-\rho^2)^2}{2k} + O\left(\frac{1}{k^2}\right)\right)$$

From here we can calculate the bias of $(g(\hat{\rho}))$:

$$\begin{aligned}
Bias(g(\hat{\rho})) &= e^{\rho-1}\left[\exp\left(O\left(\frac{1}{k}\right)\right)\left(1+\frac{(1-\rho^2)^2}{2k}+O\left(\frac{1}{k^2}\right)\right)-1\right] \\
&= e^{\rho-1}\left[\exp\left(O\left(\frac{1}{k}\right)\right)-1\right]+e^{\rho-1}\left[\exp\left(O\left(\frac{1}{k}\right)\right)\cdot\left(\frac{(1-\rho^2)^2}{2k}+O\left(\frac{1}{k^2}\right)\right)\right] \\
&\leq e^{\rho-1}\left[O\left(\frac{1}{k}\right)\cdot\exp\left(O\left(\frac{1}{k}\right)\right)\right]+e^{\rho-1}\cdot O\left(\frac{1}{k}\right), \text{ since } |e^x-1|\leq |x|e^{|x|} \\
&= e^{\rho-1}\left[O\left(\frac{1}{k}\right)\cdot\exp\left(O\left(\frac{1}{k}\right)\right)+O\left(\frac{1}{k}\right)\right] \\
&= O\left(\frac{1}{k}\right), \text{ completing the proof}
\end{aligned}$$

□

4.2.2 Variance of $g(\hat{\rho}_{mult})$

Theorem 4.2.2. *For the non-quantized multiplicative estimator,*

$Var(f(\hat{\rho})) = \frac{(1-\rho^2)^2\exp\left(\frac{2(\rho-1)}{\sigma^2}\right)}{k\cdot\sigma^4} = O(\frac{1}{k})$

Proof. We can create an upper bound on $\mathrm{Var}(f(\hat{\rho}))$ via Taylor expansion, again using the Lagrange form of the remainder:

$$\begin{aligned}
\mathrm{Var}(g(\hat{\rho})) &= \mathrm{Var}\left[g(E(\hat{\rho}))+g'(E(\hat{\rho}))(\hat{\rho}-E(\hat{\rho}))+g''(\tilde{\rho})\Big(\hat{\rho}-E(\hat{\rho})\Big)^2\right] \\
&= 0+\mathrm{Var}(\hat{\rho})(g'(E(\hat{\rho}))^2+R \\
&= \Big(\frac{(1-\rho^2)^2}{k}+O\left(\frac{1}{k^2}\right)\Big)\left(\frac{\exp((\rho-1+O\left(\frac{1}{k}\right))/\sigma^2)}{\sigma^2}\right)^2+R \\
&= \frac{(1-\rho^2)^2\exp\left(\frac{2(\rho-1)}{\sigma^2}\right)}{k\cdot\sigma^4}+R
\end{aligned}$$

Thus the first order term is $O\left(\frac{1}{k}\right)$ with a constant in terms of ρ and σ. We now look at remainder term $R's$ individual components to show that they are of lower order than $O\left(\frac{1}{k}\right)$:

$$R = \left(\frac{g''(E(\tilde{\rho}))^2}{4}\right) \mathrm{Var}\Big[(\hat{\rho} - E(\hat{\rho}))^2\Big] +$$

$$+ \mathrm{Cov}\Big((\hat{\rho} - E(\hat{\rho}), (\hat{\rho} - E(\hat{\rho}))^2\Big) g'(E(\hat{\rho})) g''(\tilde{\rho})$$

□

Lemma 4.2.3. *Var*$[(\hat{\rho} - E(\hat{\rho}))^2] = O(\frac{1}{k^2})$ *and* *Cov*$\Big((\hat{\rho} - E(\hat{\rho}), (\hat{\rho} - E(\hat{\rho}))^2\Big) = O(\frac{1}{k^{1.5}})$.

Proof. We use the same sub-exponential argument as in (4.1.3). We note that

$$\forall i, E(z_i, z_i') = \mathrm{Cov}(z_i, z_i') + E(z_i)E(z_i') = \rho$$

and so $(z_i z_i' - \rho)$ is sub-exponential with constant C. We then form sub-exponential random variable ζ_i as in:

$$\sum_i^k \zeta_i \equiv \hat{\rho} - \rho = \sum_i^k (z_i z_i' - \rho)/k$$

where ζ_i is itself sub-exponential with constant $C/\sqrt{k}$, and so

$$E[|V|^q] \leq (C'q)^q \text{ for some } C' = \alpha C, \alpha > 0$$

And so it follows that $E(\mid \hat{\rho} - \rho \mid^4) = O(k^{-2})$ and $E(\mid \hat{\rho} - \rho \mid^3) = O(k^{-1.5})$ □

Completion of Proof for Var$(g(\hat{\rho}))$

We now have that

$$\begin{aligned}
\text{Var}(g(\hat{\rho})) &= \frac{(1-\rho^2)^2 \exp\left(\frac{2(\rho-1)}{\sigma^2}\right)}{k \cdot \sigma^4} + O\left(\frac{1}{k^{3/2}}\right) + O\left(\frac{1}{k^{3/2}}\right) \\
&= \frac{(1-\rho^2)^2 \exp\left(2(\rho-1)\right)}{k} \text{ when } \sigma = 1 \\
&= O\left(\frac{1}{k}\right), \text{ as required}
\end{aligned}$$

We plot the theoretical bounds calculated above against simulated values for emphasis.

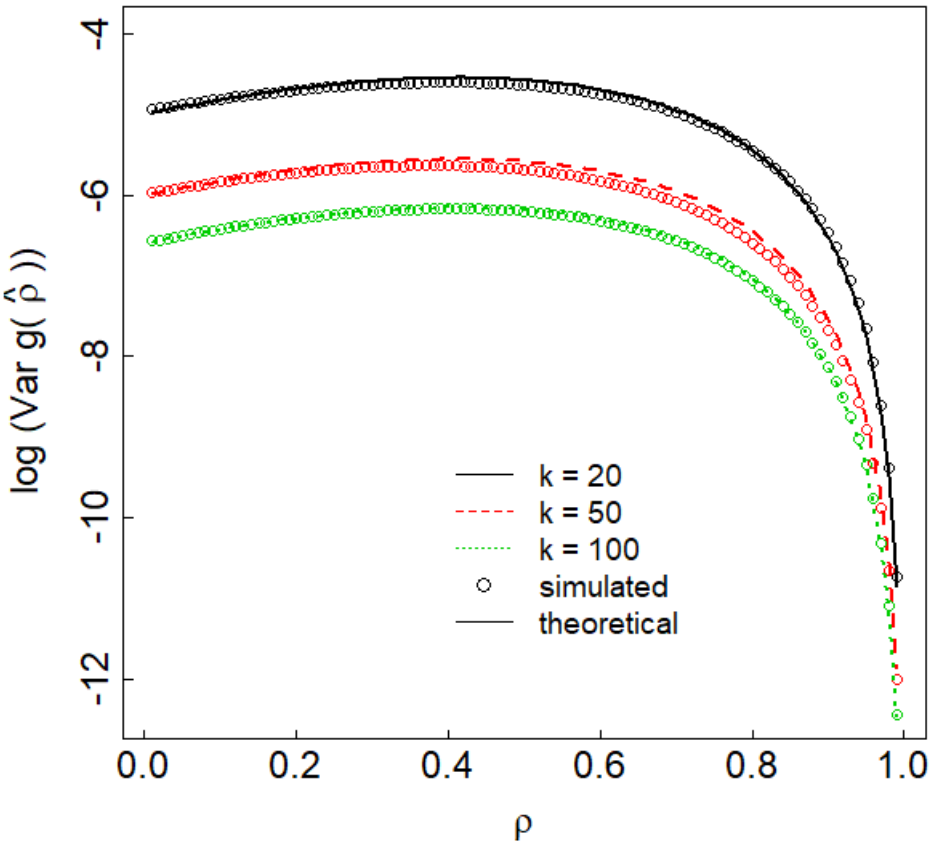


Figure 4.3: Plot of log $\mathrm{Var}(g(\rho_{mult}))$ against ρ, simulated and theoretical; we plot log values so that differences are clearer. The close fit shows that the true values of $\mathrm{Var}(g(\hat{\rho}_{mult}))$ helps confirm our calculated $O\left(\frac{1}{k}\right)$.

4.3 Bounds for Gaussian measure $g(\hat{\rho})$ for quantized $\hat{\rho}$

In this section we show that the MSE for the Gaussian distance measure using quantized estimators, both deterministic and stochastic, is also of order $O\left(\frac{1}{k}\right)$. We start with Deterministic Quantization then move on to Stochastic.

4.3.1 Delta method for expectation $g(\hat{\rho}_{q_M})$, 1-bit Quantized MLE

Theorem 4.3.1. *For 1-bit Quantized MLE, bias* $(g(\hat{\rho})) = O(\frac{1}{k})$

Proof. The methodology for this proof is the delta method similar to the previous section, using the $O\left(\frac{1}{k}\right)$ bias of $\hat{\rho}$ established above. Now recall that, for the Quantized MLE $\hat{\rho}_{q_M}$:

$$E[\hat{\rho}_{q_M}] = \rho + \frac{\rho}{2k}\left(\arcsin^2(\rho) - \frac{\pi^2}{4}\right)$$

$$Var(\hat{\rho}_{q_M}) = \frac{1}{k}(1-\rho^2)\left(\frac{\pi^2}{4} - \arcsin^2(\rho)\right) + O\left(\frac{1}{k^2}\right)$$

Then taking a Taylor expansion, we have as before

$$E(g(\hat{\rho})) = g(E(\hat{\rho})) + g''(E(\tilde{\rho}))\mathrm{Var}(\hat{\rho})$$

$$\begin{aligned}
\text{So } E(g(\hat{\rho})) &= \exp\left(\rho + O\left(\frac{1}{k}\right) - 1\right) \cdot (1 + \mathrm{Var}(\hat{\rho})) \\
&= \exp\left(\rho + O\left(\frac{1}{k}\right) - 1\right) \cdot \left(1 + \frac{(1-\rho^2)}{k} \cdot \left(\frac{\pi^2}{4} - \arcsin^2(\rho)\right) + O(\frac{1}{k^2})\right) \\
&= \exp(\rho - 1) \cdot \exp\left(O\left(\frac{1}{k}\right)\right) \cdot \left(1 + \frac{(1-\rho^2)}{k} \cdot \left(\frac{\pi^2}{4} - \arcsin^2(\rho)\right) + O\left(\frac{1}{k^2}\right)\right)
\end{aligned}$$

We can now calculate $bias(g(\hat{\rho}))$ by subtracting the above from $g(\hat{\rho})$:

$$bias(g(\hat{\rho})) = e^{\rho-1}\left[\exp\left(O\left(\frac{1}{k}\right)\right)\left(1+\frac{(1-\rho^2)}{k}\cdot\left(\frac{\pi^2}{4}-\arcsin^2(\rho)\right)+O\left(\frac{1}{k^2}\right)\right)-1\right] \tag{4.7}$$

$$= e^{\rho-1}\left[\exp\left(O\left(\frac{1}{k}\right)\right)-1\right]+ \tag{4.8}$$

$$+ e^{\rho-1}\left[\exp\left(O\left(\frac{1}{k}\right)\right)\cdot\left(\frac{(1-\rho^2)}{k}\cdot\left(\frac{\pi^2}{4}-\arcsin^2(\rho)\right)+O\left(\frac{1}{k^2}\right)\right)\right] \tag{4.9}$$

$$\leq e^{\rho-1}\left[O\left(\frac{1}{k}\right)\cdot\exp\left(O\left(\frac{1}{k}\right)\right)\right]+e^{\rho-1}\cdot O\left(\frac{1}{k}\right),\ \text{since } |e^x-1|\leq |x|e^{|x|} \tag{4.10}$$

$$= e^{\rho-1}\left[O\left(\frac{1}{k}\right)\cdot\exp\left(O\left(\frac{1}{k}\right)\right)+O\left(\frac{1}{k}\right)\right] \tag{4.11}$$

$$= O\left(\frac{1}{k}\right),\ \text{completing the proof} \tag{4.12}$$

□

4.3.2 Delta method / Taylor expansion for expectation $g(\hat{\rho}_{qS})$, Stochastic Quantization estimator

Theorem 4.3.2. *For the Stochastic Quantized estimator, Bias* $(g(\hat{\rho}_{qS})) = O(\frac{1}{k})$

Proof. The proof for SQ is similar, as the step in line (4.7) holds for any $O\left(\frac{1}{k}\right)$ variance.

Recall that, for the one-dimensional Stochastic Quantized estimator with m levels:

$$E[\hat{\rho}_{q_S}] = \rho$$

$$Var(\hat{\rho}_{q_S}) = \sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot \frac{|Z - B_i|}{d} \cdot \frac{|Z' - B_j|}{d} \cdot$$

$$P(Z \in (B_{i-1}, B_{i+1}), Z' \in (B_{j-1}, B_{j+1})) - \rho^2$$

Thus, in quantizing a k-dimensional random projection, our estimator is $\langle Q, Q' \rangle = \sum_l^k \frac{1}{k} q_l q_l'$ which has variance:

$$Var(\hat{\rho}_{q_S}) = \sum_{l}^{k} \frac{1}{k} \Big[\sum_{i=1}^{m}\sum_{j=1}^{m} B_i^2 B_j^2 \cdot \frac{|Z_l - B_i|}{d} \cdot \frac{|Z_l' - B_j|}{d}$$

$$P(Z_l \in (B_{i-1}, B_{i+1}), Z_l' \in (B_{j-1}, B_{j+1})) - \rho^2 \Big]$$

And so we can substitute this in to our Taylor expansion above:

$$E(g(\hat{\rho})) = g(E(\hat{\rho})) + g''(E(\hat{\rho}))\mathrm{Var}(\hat{\rho})$$

$$\text{So } E(g(\hat{\rho})) = \exp(\rho - 1) \cdot (1 + \mathrm{Var}(\hat{\rho}))$$

$$= \exp(\rho - 1) \cdot \left(1 + O\left(\frac{1}{k}\right) + O\left(\frac{1}{k^2}\right)\right)$$

Similarly to above,

$$\text{Bias } g(\hat{\rho}) = \exp(\rho - 1) \cdot \left[\left(1 + O\left(\frac{1}{k}\right) + O\left(\frac{1}{k^2}\right)\right) - 1\right]$$

$$= exp(\rho - 1)\left(O\left(\frac{1}{k}\right) + O\left(\frac{1}{k^2}\right)\right)$$

$$= O\left(\frac{1}{k}\right)$$

□

4.3.3 Bounds for $g(\hat{\rho}_{lin})$, general b bit quantized estimator

Variance and Bias for $\hat{\rho}_{lin}$

We begin by noting that, from [7] and some previous work, we have the following bounds for $\hat{\rho}_{lin} = \frac{1}{k}\langle q, q' \rangle$:

$$\text{Var}(\hat{\rho}_{lin}) \leq \frac{1+\rho^2}{k} \tag{4.13}$$

$$\text{Bias}^2(\hat{\rho}_{lin}) \leq 4\rho^2 D_b^2, \text{ where } D_b = \frac{3^{1.5}2\pi}{12}2^{-2b} \tag{4.14}$$

For simplicity we call this

$$\text{Bias}^2(\hat{\rho}_{lin}) \leq \rho^2 c^2 \tag{4.15}$$

Note that $\text{Var}(\rho_{lin})$ is increasing as b increases, whereas bias is independent of k. As noted in [7], concerning estimation of ρ, when considering the trade off between b and k (on the assumption that we have a total of $B = b \cdot k$ bits), there is a trade off of variance and bias as k and b are changed. While variance does increase somewhat as b increases, bias is completely independent of k and only depends on b and ρ.

Theorem 4.3.3. $|E(g(\hat{\rho}))| \leq |e^{\rho-1} \cdot (1 - e^{\sqrt{3}\pi\rho 2^{-2b}})|$

Proof. We assume as usual that $\rho \geq 0$. Then we have

$$|\text{Bias}(\hat{\rho})| \leq c\rho, -c\rho \leq \text{Bias}(\hat{\rho}) \leq c\rho$$

$$|\text{Bias}(\hat{\rho})| \equiv |E(\hat{\rho}) - \rho|$$

$$|E(\hat{\rho}) - \rho| \leq \rho c$$

$$\rho(1-c) \leq E[\hat{\rho}] \leq \rho(1+c)$$

We then take the Taylor expansion. We use the Lagrange form of the remainder, with $\tilde{\rho} \in (\rho, \hat{\rho})$, and so $\tilde{\rho} \leq \max(\rho, \hat{\rho})$.

$$E[g(\hat{\rho}] = g(E[\hat{\rho}]) + \frac{1}{2}g''(\tilde{\rho})\text{Var}(\hat{\rho})$$

The individual terms can be broken down:

$$g(E[\hat{\rho}]) = e^{E[\hat{\rho}]-1}$$

$$e^{\rho(1-c)-1} \leq e^{E[\hat{\rho}]-1} \leq e^{\rho(1+c)-1}$$

The upper bound using the remainder and (4.13) is straightforward:

$$\frac{1}{2}g''(\tilde{\rho}) \leq \frac{1}{2}e^{\rho(1+c)-1}\left(\frac{1+\rho^2}{k}\right)$$

Which gives us an upper bound for the second term in $\text{Var}(\hat{\rho})$ since $g''(\cdot)$ is positive. We take these knowns and apply delta method to upper bound $E[g(\hat{\rho})]$.

Now, for absolute bias of $(g(\hat{\rho}))$, we take upper and lower bounds of :

$$E(g(\hat{\rho})) = g(E(\hat{\rho})) + g''(E(\tilde{\rho}))\text{Var}(\hat{\rho})$$

$$E(g(\hat{\rho})) \leq \exp(\rho - 1)\exp(|c2^{-2b}|) + R,$$

$$E(g(\hat{\rho})) \geq \exp(\rho - 1)\exp(-|c2^{-2b}|) + R$$

Now focusing on remainder R and setting $\sigma = 1$:

$$R = \exp(E(\tilde{\rho}) - 1)) \cdot \frac{1+\rho^2}{k}$$

$$\leq \exp(E(\max(\rho, \hat{\rho}) - 1)) \cdot \frac{1+\rho^2}{k}$$

Bias becomes squared in forming the MSE, so we can safely drop the $O(\frac{1}{k})$ since it will disappear rapidly next to the Variance term (which is itself $O(\frac{1}{k})$, as we will explore shortly). Doing so, we now take absolute value of bias of $g(\hat{\rho})$:

$$\begin{aligned}
|E(g(\hat{\rho})) - g(\rho))| &= e^{\rho-1} \cdot |(e^{\text{bias}(\hat{\rho})} - 1)| \\
&\leq min\Big(e^{\rho-1} \cdot |e^{|\text{bias}(\hat{\rho})|} - 1|, e^{\rho-1} \cdot |1 - e^{-|\text{bias}(\hat{\rho})|}|\Big) \\
&\leq min\Big(e^{\rho-1} \cdot (e^{|E[\hat{\rho}]-\rho|} - 1), e^{\rho-1} \cdot (1 - e^{-|E[\hat{\rho}]-\rho|})\Big)
\end{aligned}$$

□

Theorem 4.3.4. *Variance of* $g(\hat{\rho}_{lin}) = \frac{(1+\rho^2)e^{2\rho}}{e^2} \cdot \frac{1}{k} \cdot \exp\left(2\pi\sqrt{3} \cdot 2^{-2b}\right) + O(k^{-1.5}) = O(1/k)$.

Proof. Variance calculation is similar:

$$\begin{aligned}
\text{Var}(g(\hat{\rho})) &= \text{Var}\left[g(E(\hat{\rho})) + g'(E(\hat{\rho}))\left((\hat{\rho}) - E(\hat{\rho})\right) + g''(\hat{\rho}))\left((\hat{\rho}) - E(\hat{\rho})\right)^2\right] \\
&= 0 + \text{Var}(\hat{\rho})\left[g'(E[\hat{\rho})]\right]^2 + R
\end{aligned}$$

We calculate the first term:

$$\begin{aligned}
\text{Var}(\hat{\rho}\left[g(\hat{\rho}))^2\right] &= \frac{1+\rho^2}{k} \exp\Big(2(\rho-1) + 2 \cdot c2^{-2b}\Big) \\
&= \frac{1+\rho^2}{k} \exp\Big(2\rho + 2\pi\sqrt{3}\rho 2^{-2b} - 2\Big) \\
&= \frac{(1+\rho^2)e^{2\rho}}{e^2} \cdot \frac{1}{k} \cdot \exp\Big(2\pi\sqrt{3} \cdot 2^{-2b}\Big)
\end{aligned}$$

Now we calculate the remainder R, which will involve several sub-lemmas.

$$R = \left(\frac{f''(E(\tilde{\rho}))^2}{4}\right) \text{Var}(\hat{\rho} - E(\hat{\rho})^2) + \text{Cov}\Big((\hat{\rho} - E(\hat{\rho}), (\hat{\rho} - E(\hat{\rho}))^2\Big) f'(E(\hat{\rho})) f''(\tilde{\rho})$$

Lemma 4.3.5. *Generally,* $E[\|\hat{\rho}-\rho\|^m] = O\left(k^{-m/2}\right)$*, for* $m \geq 1$*. And so* $Var[(\hat{\rho}-E(\hat{\rho}))^2] = O(\frac{1}{k^2})$.

Proof. Note that this is the same basic proof used for the non-quantized linear estimator, and hinges on basic facts: each pair (q_l, q_l') are i.i.d., and both $\mathrm{E}(q_l q_l')$ and $\mathrm{Var}(q_l q_l') \equiv \sigma_l^2$ are finite since each q_l, q_l' have bounded, discrete values.

$$\forall i, E(z_i, z_i') = \mathrm{Cov}(z_i, z_i') + E(z_i)E(z_i') = \rho$$

We note that $\hat{\rho}_{lin} = \frac{1}{k}\langle q, q' \rangle$ is sub-exponential. The proof is then identical to that of (4.1.3). □

MSE of $g(\hat{\rho}_{lin})$

Putting the MSE together, we have, for general b-bit linear estimator:

$$\mathrm{MSE}(g(\hat{\rho})) = \mathrm{Bias}^2(g(\hat{\rho})) + \mathrm{Var}(g(\hat{\rho})) \tag{4.16}$$

$$= \left\{ e^{\rho-1} \exp(c2^{-2b}) \left[\left(c2^{-2b} \right) + \frac{1+\rho^2}{k} \right] \right\}^2 + \frac{(1+\rho^2)e^{2\rho}}{k \cdot e^2} \cdot \exp\left(2\pi\sqrt{3} \cdot 2^{-2b} \right) \tag{4.17}$$

$$= \frac{e^{2\rho}}{e^2} \exp(2\pi\sqrt{3} \cdot 2^{-2b}) \left[\left(c2^{-2b} \right) + \frac{1+\rho^2}{k} \right]^2 + \frac{(1+\rho^2)e^{2\rho}}{k \cdot e^2} \cdot \exp\left(2\pi\sqrt{3} \cdot 2^{-2b} \right) \tag{4.18}$$

$$= e^{2(\rho-1)} \exp\left(2\pi\sqrt{3} \cdot 2^{-2b} \right) \cdot \left\{ \left[\left(c2^{-2b} \right) + \frac{1+\rho^2}{k} \right]^2 + \frac{1+\rho^2}{k} \right\} \tag{4.19}$$

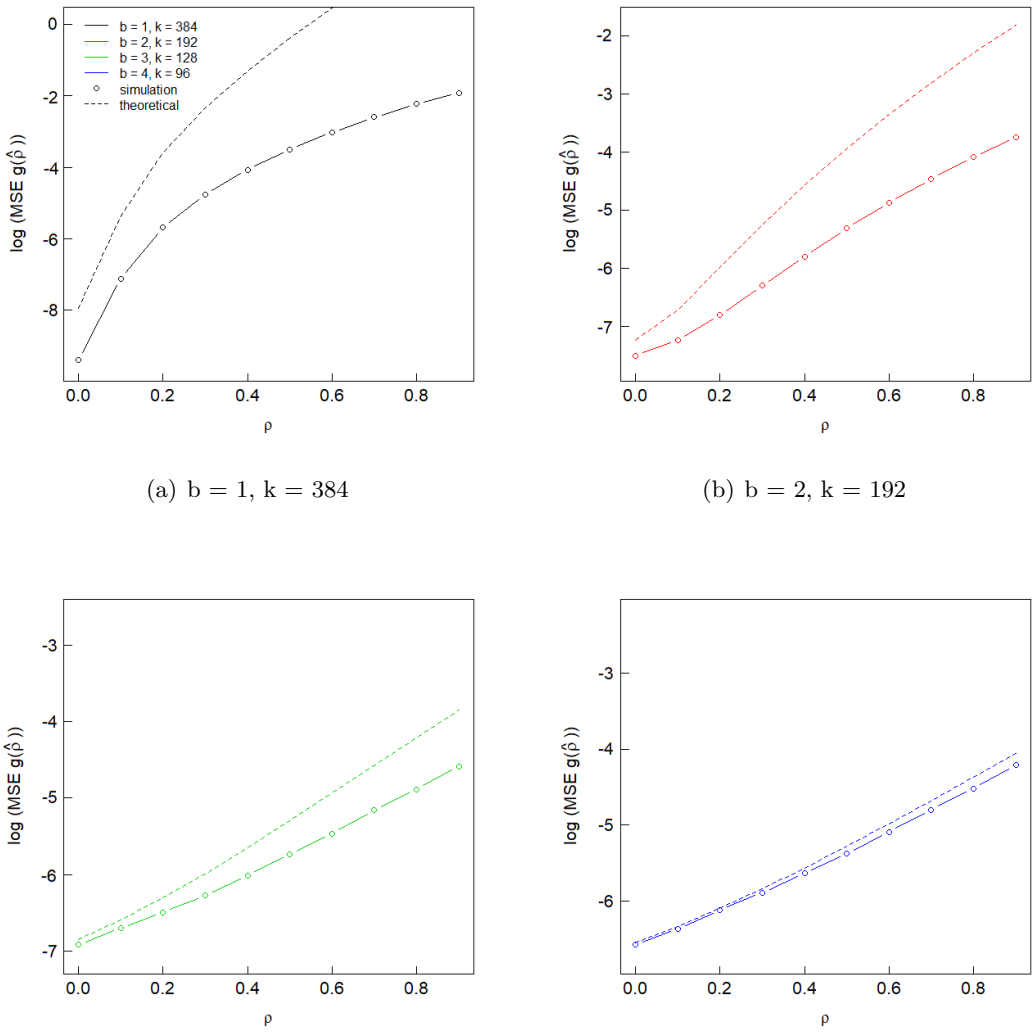
□

Plots using numerical integration

As the above form is rather complex, we plot lines of these calculated bounds along with simulated MSE of $\hat{\rho}$ in Fig. 4.19. We fix the total number of bits used in compression B, and plot several values of b and k (i.e. $B = b \cdot k$ is the same for all levels).

4.4 Discussion

In this chapter we established theoretical bounds on estimation of the Guassian similarity measure $g(\rho) = \exp(\rho - 1)/\sigma^2$, used in a variety of applications but in particular spectral clustering. We show that quantized random projection does preserve $g(\rho)$, with MSE of order $\left(\frac{1}{k}\right)$, and is thus suitable for situations where similar functions of pairwise distance are involved. Having worked out the theory, we are now prepared to move on to experiments on both synthetic and real datasets.



(a) b = 1, k = 384

(b) b = 2, k = 192

(c) b = 3, k = 128

(d) b = 4, k = 96

Figure 4.4: log MSE of $\hat{\rho}$, quantized linear estimator. We vary levels of $b \cdot k = 384$. Theoretical upper bound as calculated above in Eq. 4.19. While this plot is mostly to compare simulated values and theoretical upper bounds, we also note that MSE improves rapidly as b increases from 1 to 3, where it starts to stablize.

Chapter 5: Spectral Clustering and Random Projection Experiments

Having established theoretical bounds of the Gaussian similarity function $g(\hat{\rho})$ in Chapter 4, we explore how spectral clustering (SC) actually performs after applying random projection. In particular, we are interested in the relationship between mean squared error (MSE) of similarity measures and clustering performance, and how the parameters of our algorithm (spectral clustering and random projection both) affect performance. This is extended to analysis of the relationship between both MSE and performance and various parameters: ρ, σ, k, as well as two different estimators $\hat{\rho}_{mult}$ and $\hat{\rho}_{lin}$. We conduct simulations under two experiment regimes: one in which elements within a block have the same correlation with all other elements in the block; and one in which each block is an autoregressive system of order 1, AR(1) for short. We also perform several real data experiments.

Chapter Outline

We start by describing our simulation experiments, the theory behind them, and results. The second section goes into details of how the various parameters of SC affect MSE ($\hat{\rho}$) and how that affects clustering results. In the third and final section we discuss our real data experiments.

5.1 Spectral Clustering Simulation Experiments

The purpose of these simulation experiments was to explore in depth how random projection works in practice, with a controlled setting. We construct two related simulation experiments wherein we can control every aspect, which we now describe in detail.

5.1.1 Simulation Setting Details

1) We assume original, pre-compressed data $x = x_1, ..., x_n$, each x_i a d-length Gaussian vector with mean 0, covariance matrix C (i.e., $x \sim N(0, C)$). The covariance matrix $C_{n \times n}$ is defined by the following form:

$$C = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}$$

That is to say, we divide our data into 2 blocks, B_1 and B_2. Elements within the same block have covariance ρ; elements not within the same block have covariance 0 (i.e., are independent).

2) We then generate simulated random projection result $Z_{n \times k}$ as a k-dimensional multivariate normal,

$$Z_i, Z_j \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \rho_{i,j} & 0 \\ 0 & \rho_{i,j} \end{pmatrix}\right)$$

3) We generate $z = Rx$, where z is the desired multivariate normal $z \sim N(\mu, \Sigma^2)$, $RR^T = \Sigma$, and $x \sim N(0, 1)$ as above. We solve for R using the following:

$$R = \begin{pmatrix} D & c/\sqrt{n} & ... & & \\ c/\sqrt{n} & D & c/\sqrt{n} & ... & \\ c/\sqrt{n} & c/\sqrt{n} & D & c/\sqrt{n} & ... \\ ... & & & & \\ c/\sqrt{n} & ... & ... & c/\sqrt{n} & D \end{pmatrix}$$

$$R \cdot R^T = \begin{pmatrix} 1 & \rho & \rho & \rho & ... \\ \rho & 1 & \rho & \rho & ... \\ \rho & \rho & 1 & \rho & ... \\ ... & & & & \\ \rho & ... & ... & \rho & 1 \end{pmatrix}$$

Which gives us two linear equations with which we can solve for c and D.

4) For the first version of this experiment (henceforth "regime 1") we have fixed ρ within all blocks as described above, so the only parameters in model are B blocks and ρ.

The second model (regime 2) is an autoregressive model of order 1, i.e. it uses power decay where $\rho_{i,j} = \rho^{|i-j|}$ within blocks and 0 between, instead of constant ρ within blocks.

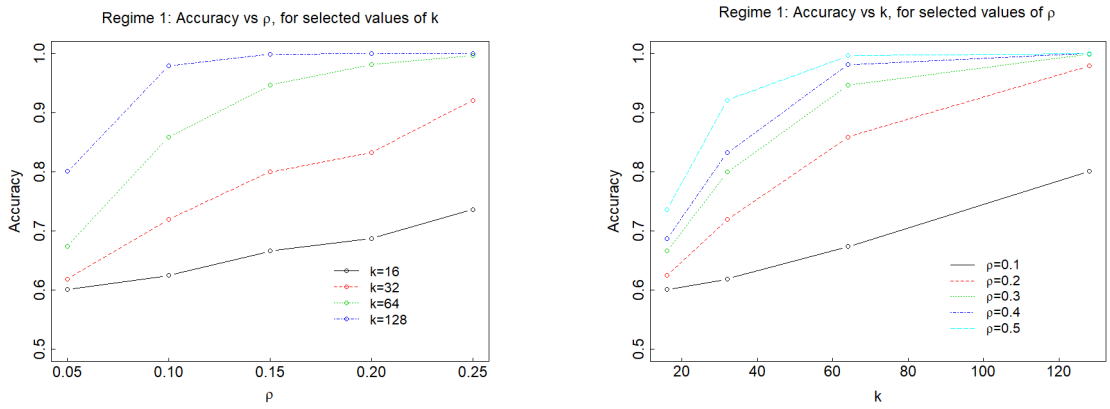
5) After generating this data Z, we calculate g($\hat{\rho}$) for Z and then perform spectral clustering as usual (see in Appendix B for full details).

5.1.2 Initial Results: Clustering Accuracy against ρ and k

We begin by displaying initial results of the regime 1 experiment, Fig. 5.1. We plot clustering accuracy of spectral clustering with random projection against both original correlation ρ and reduced dimension k to show the basic outline of our results. As we expected, performance increases very clearly with both ρ and k.

Remark: Note on defining clustering accuracy

Since there are only two clusters, we can define accuracy simply as the proportion of observations in the same (true) block that are in the same cluster after SCRP. Thus, clustering accuracy of 1 indicates that all observations in a (true) block are clustered together, whereas the minimum clustering accuracy of 0.5 indicates that exactly half of the observations are



(a): Accuracy by ρ (selected k)

(b): Accuracy by k (selected ρ)

Figure 5.1: Clustering Accuracy, Regime 1, n = 1000, replicates = 100, $\sigma = 1$. These two plots are the initial results using the linear estimator $\hat{\rho}_{lin}$, demonstrating the relationship of Accuracy with ρ (true, original correlation) and k (reduced dimensionality).

in an incorrect cluster.

5.1.3 Linear versus Multiplicative Estimator

In this subsection we explore the linear and multiplicative estimators $\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$. We begin with a discussion of theoretical bounds on MSE then discuss regime 1 results comparing $\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$.

$\hat{\rho}_{lin}$ known quantities, from previous work

$$g(\rho) = \exp\left(\frac{\rho}{\sigma^2}\right) \qquad g'(\rho) \ = \frac{1}{\sigma^2}\exp\left(\frac{\rho}{\sigma^2}\right) \qquad g''(\rho) \ = \frac{1}{\sigma^4}\exp\left(\frac{\rho}{\sigma^2}\right)$$

$$\text{Bias}(\hat{\rho}) \leq \frac{\exp\left(\rho/\sigma^2\right)\cdot(1+\rho^2)}{k\cdot\sigma^4}$$

$$\begin{aligned}\text{Var}(g(\hat{\rho})) &= \text{Var}(\hat{\rho})(g'(E(\hat{\rho}))^2 + \text{remainder} \\ &\leq \frac{\exp\left(2\rho/\sigma^2\right)\cdot(1+\rho^2)}{k\cdot\sigma^4} + o\left(\frac{1}{k}\right)\end{aligned}$$

$\hat{\rho}_{mult}$ known quantities, from previous work

$$\text{Bias}(\hat{\rho}) \leq \frac{5\rho+\rho^3/2}{k}$$

$$\text{Var}(\hat{\rho}) = \frac{(1-\rho^2)}{k} + o\left(\frac{1}{k}\right)$$

Calculation for multiplicative estimator

$$E(g(\hat{\rho})) = \exp\left(\frac{\rho+\text{Bias}(\hat{\rho})}{\sigma^2}\right)\left(1+\frac{1}{\sigma^4}\cdot\text{Var}(\hat{\rho})\right), \text{ we drop the } o\left(\frac{1}{k}\right) \text{ term in } \text{Var}(\hat{\rho}) \tag{5.1}$$

$$= \exp\left(\frac{\rho}{\sigma^2}\right)\cdot\exp\left(\frac{\text{Bias}(\hat{\rho})}{\sigma^2}\right)\cdot\left(1+\frac{1}{\sigma^4}\cdot\frac{(1-\rho^2)^2}{k}\right) \tag{5.2}$$

$$= \exp\left(\frac{\rho}{\sigma^2}\right)\cdot\exp\left(\frac{5\rho+\rho^3/2}{k\sigma^2}\right)\cdot\left(1+\frac{1}{\sigma^4}\cdot\frac{(1-\rho^2)^2}{k}\right) \tag{5.3}$$

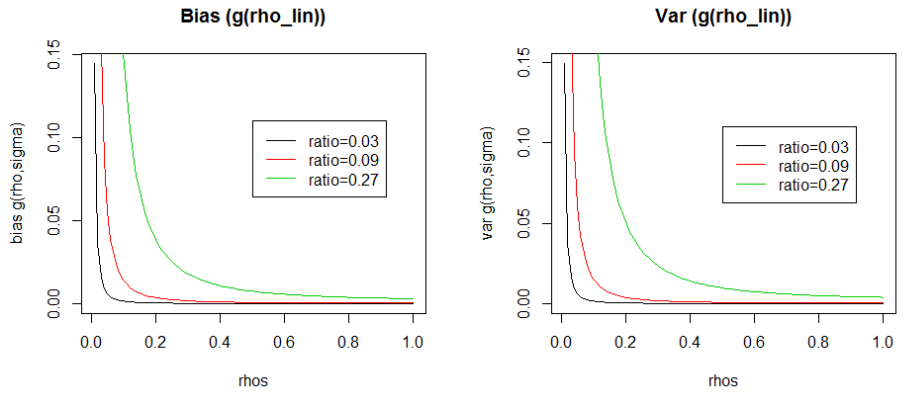
Thus we can calculate bias by subtracting $g(\rho)$ from 5.3:

$$\begin{aligned}
\text{Bias} &= \exp\left(\frac{\rho}{\sigma^2}\right) \cdot \exp\left(\frac{5\rho + \rho^3/2}{k\sigma^2}\right) \cdot \left(1 + \frac{1}{\sigma^4} \cdot \frac{(1-\rho^2)^2}{k}\right) - \exp\left(\frac{\rho}{\sigma^2}\right) \\
&= \exp\left(\frac{\rho}{\sigma^2}\right) \cdot \left[\exp\left(\frac{5\rho + \rho^3/2}{k\sigma^2}\right) \cdot \left(1 + \frac{1}{\sigma^4} \cdot \frac{(1-\rho^2)^2}{k}\right) - 1\right] \\
&= O\left(\frac{1}{k}\right)
\end{aligned}$$

From previous work,

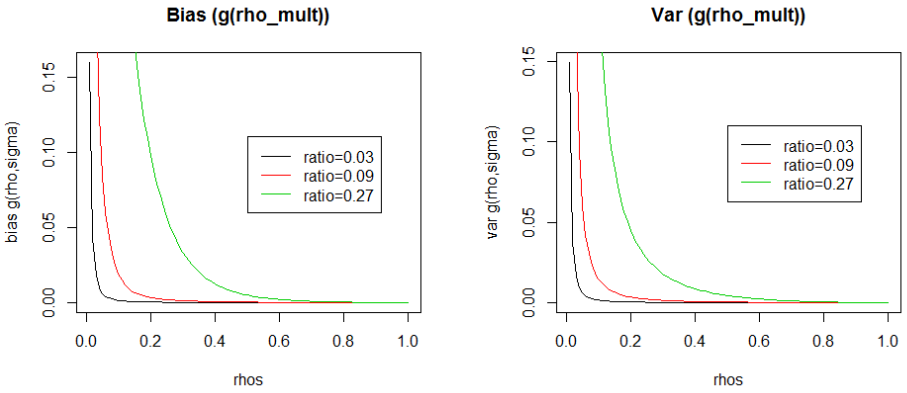
$$\text{Var}(g(\hat{\rho})) = \frac{(1-\rho^2)^2 \exp\left(\frac{2\rho}{\sigma^2}\right)}{k \cdot \sigma^4} + o\left(\frac{1}{k}\right)$$

We plot out these theoretical values of bias and variance of the above, Fig. 5.2. In summary, both have similar $O\left(\frac{1}{k}\right)$ bounds on MSE and have comparable computation time. It then follows that we must explore the actual performance of these estimators.



(a) Bias($g(\hat{\rho}_{lin})$)

(b) Var($g(\hat{\rho}_{lin})$)



(c) Bias($g(\hat{\rho}_{mult})$)

(d) Var($g(\hat{\rho}_{mult})$)

Figure 5.2: Bias and variance of $g(\hat{\rho})$ for both linear and multiplicative estimators

$\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$ comparison simulations

In Fig. 5.3 we present results for $\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$ in the form of plots of Accuracy vs ρ for both estimators and across four values of k.

It is immediately clear that both are performing similarly, and also that ρ_{lin} appears to be performing slightly better than ρ_{mult}. We take a moment to note that there is very high variance of clustering performance, which we discuss below. Plot 5.5 shows Accuracy vs ρ with standard errors for $\hat{\rho}_{lin}$; we discuss this variance in the next section. (The corresponding plot for $\hat{\rho}_{mult}$ is almost identical and thus omitted.)

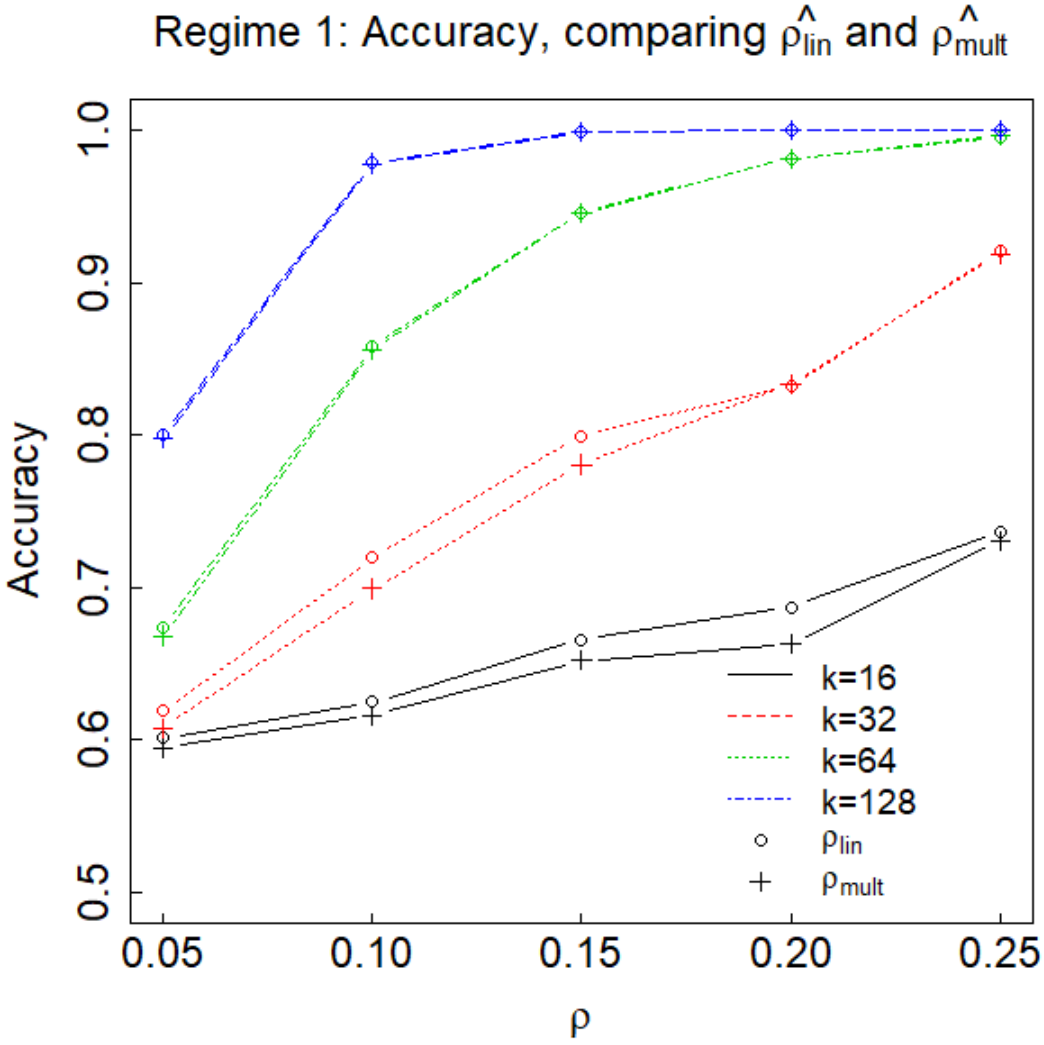


Figure 5.3: Clustering accuracy, Regime 1, n = 2000, replicates = 100. Besides the relationship between Accuracy and both ρ and k, this plot shows that the two estimators $\hat{\rho}_{lin}$ and $\hat{\rho}_{mult}$ perform similarly, with $\hat{\rho}_{lin}$ performing slightly and consistently better.

Since all our experiments suggest that $\hat{\rho}_{mult}$ and $\hat{\rho}_{lin}$ perform very similarly, we focus on $\hat{\rho}_{lin}$ in later sections.

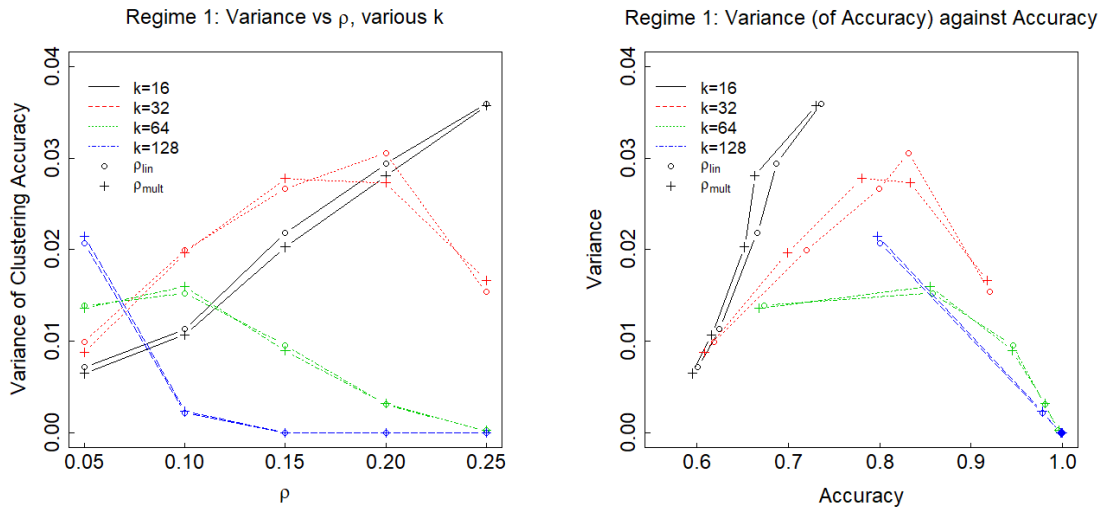


Figure 5.4: Variance of accuracy, Regime 1, n = 1000, replicates = 100, $\hat{\rho}_{lin}$. The goal of these plots is to highlight how clustering accuracy varies more when accuracy is in the "middle" range of around 0.65 to 0.85 (noting that Accuracy is between 0.5 and 1.0 for two clusters).

5.1.4 Variance of clustering accuracy

Before moving on to exploring algorithm parameters, we take a moment to note that with these relatively small sample sizes and dimensions, performance of clustering varies significantly, which explains much of the variance in the results. We plot results with standard error bars to emphasize this in Fig. 5.5.

The most important factor determining variance of accuracy is the range of accuracy itself; for example, when performance is near 100% then variance of accuracy approaches 0, and when it is near 50% there is less variance as the results are consistently near-minimum. This can be seen in Fig. 5.4. The corresponding plot for $\hat{\rho}_{mult}$ is omitted as it is essentially identical.

5.1.5 Ratios and Differences of Similarities

The spectral clustering algorithm can be viewed as clustering based on the similarity $g(\rho)$, and thus being able to distinguish $g(\rho)$ for two points within a block vs between blocks

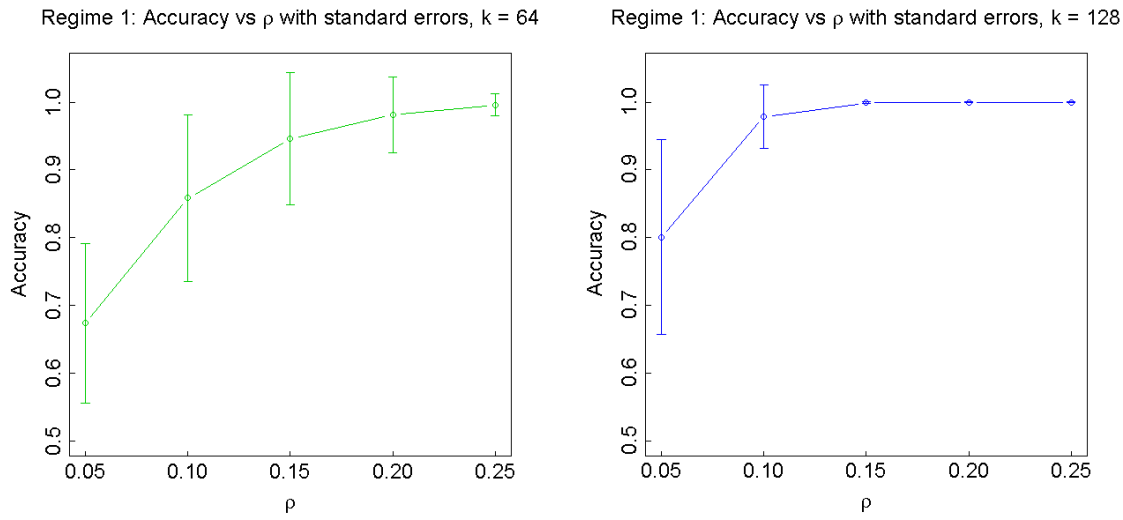


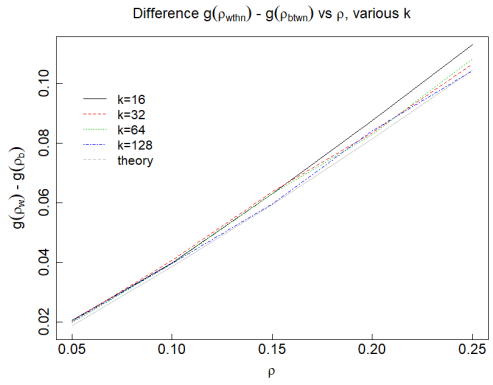
Figure 5.5: Variance of accuracy, Regime 1, n = 1000, replicates = 100, $\hat{\rho}_{lin}$. These are two of the plots we earlier combined in Fig. 5.3, with standard error bars to demonstrate how much accuracy can vary.

is potentially of interest. To explore this, we generated plots of accuracy along with ratio and difference of distances. We denote the correlation within blocks as $\rho_w := \rho$, and note that correlation between blocks $\rho_b = 0$. Of course, after random projection, the correlation between compressed data points depend on estimators $\hat{\rho}_b$ and $\hat{\rho_w}$. Then theoretical results are, respectively,

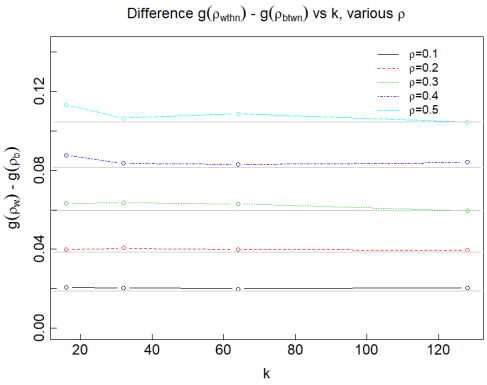
$$\text{Diff} := \exp \frac{\rho - 1}{\sigma^2} - \exp \frac{-1}{\sigma^2}$$

$$\text{Ratio} := \frac{\exp \frac{\rho - 1}{\sigma^2}}{\exp \frac{-1}{\sigma^2}}$$

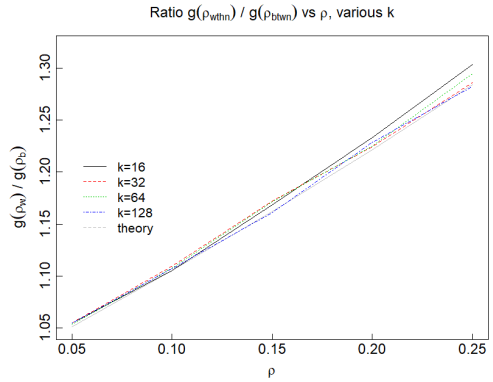
We present plots of post-random projection results against ρ and k, overlaid with the theoretical (true) values. In a later section we discuss these ratio and differences with respect to Gaussian similarity measure parameter σ.



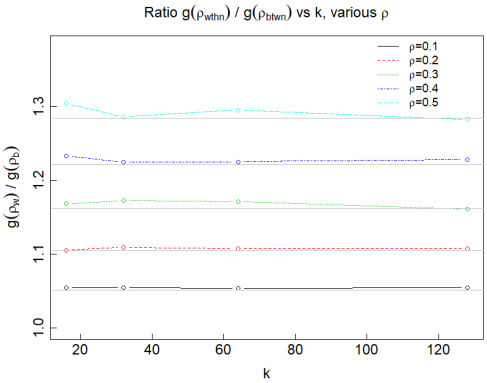
(a): Difference of g(ρ_w) - g(ρ_b) vs ρ.



(b): Difference of g(ρ_w) - g(ρ_b) vs k.



(c): Ratio of g(ρ_w) / g(ρ_b) vs ρ.



(d): Ratio of g(ρ_w) / g(ρ_b) vs k.

Figure 5.6: Difference and Ratio of g($\hat{\rho_w}$) / g($\hat{\rho_b}$), theoretical and simulated results. The aim of the ratio and difference plots is to show how separation in $g(\hat{\rho}) = \exp((\hat{\rho} - 1)/\sigma^2)$ varies with k and (true, pre-compression) ρ. The expected relationship with ρ is clear. We can also see, as k increases, the simulated plots follow the theoretical lines more closely.

5.2 Spectral Clustering and Random Projection Algorithm Parameters

In this section we explore the effects of σ, ρ, k on MSE and performance based on our experiments, as well as theory where relevant.

Parameters and quantities of Interest	
Parameter	Description
ρ	correlation between original data points, within block
k	reduced dimensionality (after random projection)
σ	σ parameter used in the Gaussian distance measure

5.2.1 Gaussian kernel parameter σ

We proceed to investigate Gaussian kernel parameter σ in detail. We begin by looking at the separation of g(ρ_b) and g(ρ_w), and plotting performance of clustering against σ.

It is evident that σ has minimal effect on performance, with the notable exception of extremely low values of σ. (In this experiment, when we dropped σ below 0, the similarity function broke down and produced Laplacians of all 0s or NaNs). Plots show that, while high sigma makes absolute weights smaller, the variance is correspondingly smaller. In Fig. 5.7 we look at separation of g(ρ_b) and g(ρ_w), i.e. ρ between and within clusters, respectively. The left panel is the difference g(ρ_b) - g(ρ_w), while the right panel is the ratio g(ρ_b)/g(ρ_w). In short, setting σ to something around 1, which other literature implicitly suggests as a default, produces good results and has the extra benefit of making calculations simpler.

We initially discussed difference and ratio in Sec. 5.1.5. We now plot theoretical within-cluster measures related to $g(\rho_b) - g(\rho_w)$, shown in Fig. 5.8, with respect to σ. In these plots we show the theoretical values of $g(\rho)$ as ρ increases, along with $g(\rho) + /- sd(\rho)$. We also define $\rho - \delta_1$, where δ_1 =absolute bias= $|\hat{\rho} - \rho|$, and within $0 - \delta_2$, then calculate and

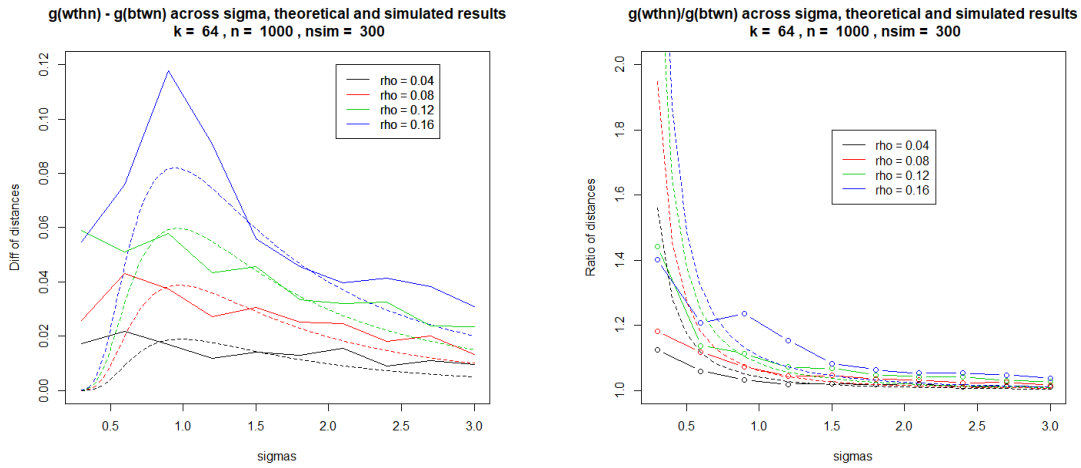


Figure 5.7: Difference of g(ρ_w) - g(ρ_b) and ratio of g(ρ_w) / g(ρ_b) against σ, theoretical and simulation results.

plot gap $= \rho - \delta_1 - \delta_2$.

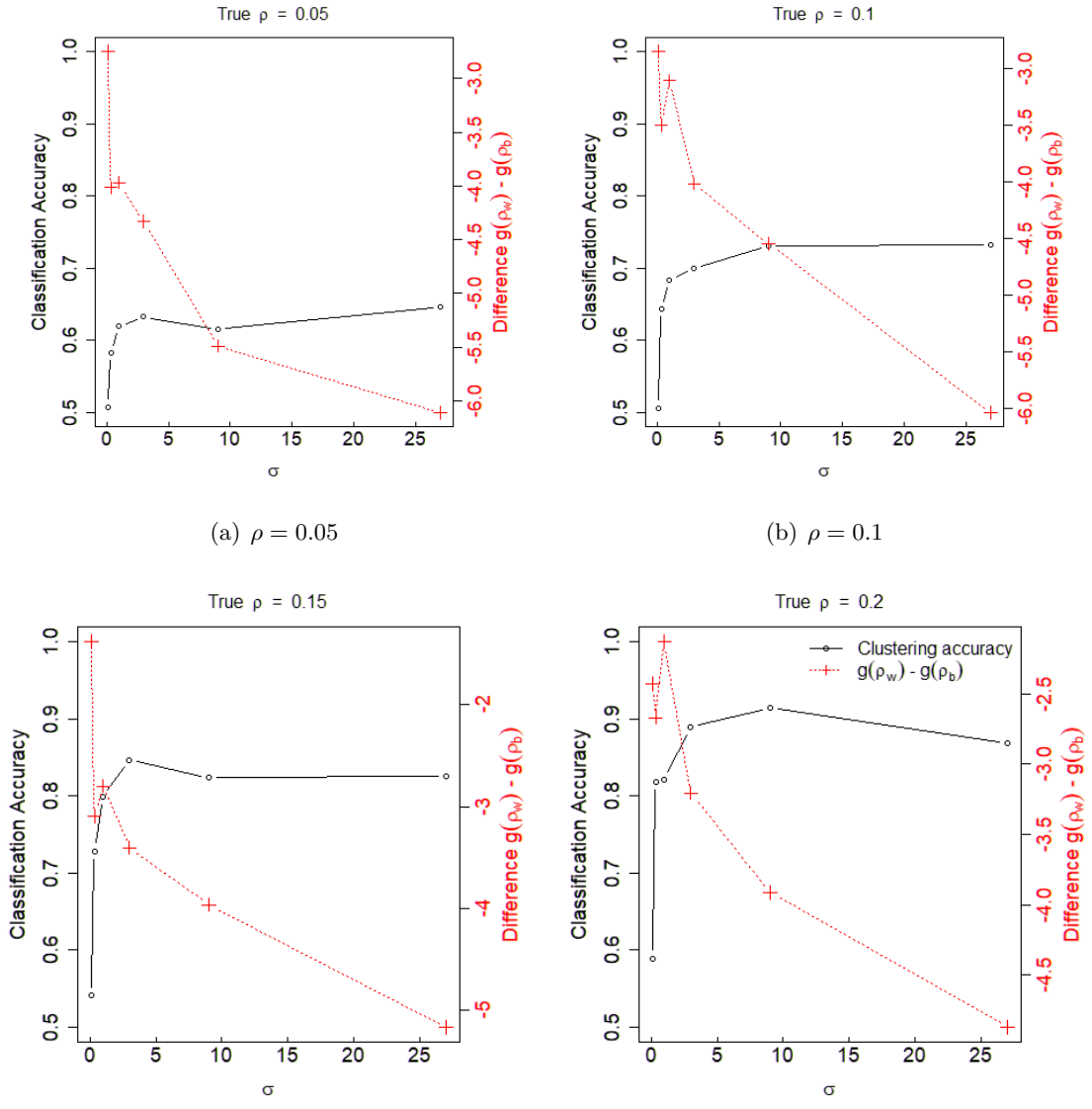
Our next plot, Fig. 5.9, examines theoretical variance under various settings. Each of the four panels in this figure has a different value of the parameter σ. The line with standard deviations crosses 1 - i.e., e^0, the point at which it would become impossible to determine which cluster our point is in - near $\rho = 0.18$ in all four plots. While there is a small amount of variation, this essentially confirms that σ should have little effect on algorithm performance.

5.2.2 Known correlation between clusters ρ

ρ has a very strong effect on performance and variance, as theory and common sense suggests. It can be seen from any of the results plots that as ρ increases, so does performance.

5.2.3 Reduced dimensionality k

As our work above suggests, MSE of ρ is $O(1/k)$, and experiments verified that there is a large effect of k on SC performance. Figures 1, 2, and 3 all clearly show a strong relationship between performance and k.



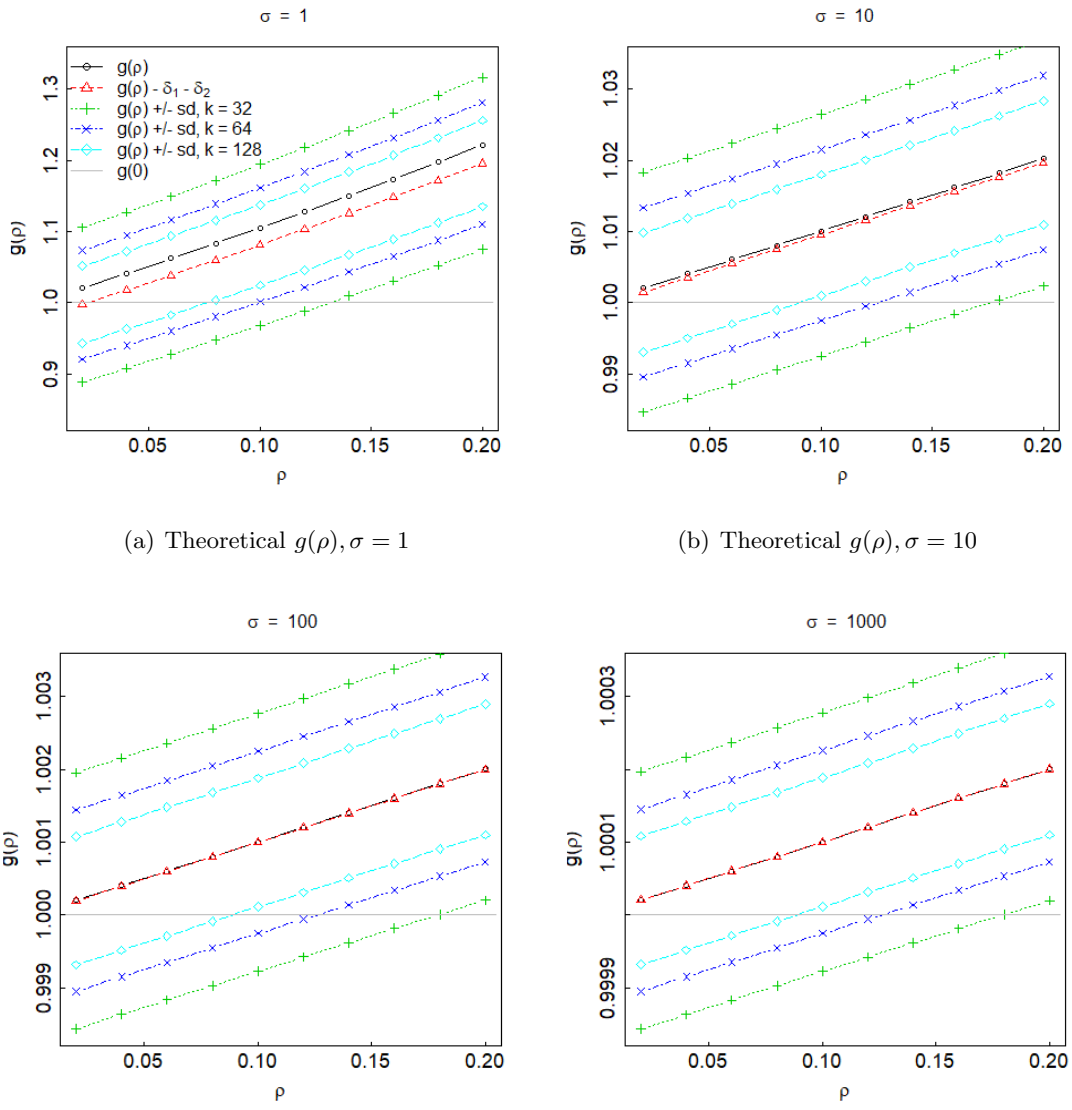
(a) $\rho = 0.05$

(b) $\rho = 0.1$

(c) $\rho = 0.15$

(d) $\rho = 0.2$

Figure 5.8: Relationship between Clustering Accuracy and Difference g(ρ_w) - g(ρ_b), across σ. Each of the four panels fixes a different value of ρ. Ultimately the plots show us that the difference in similarity measures does not tell the entire story.



(a) Theoretical $g(\rho), \sigma = 1$

(b) Theoretical $g(\rho), \sigma = 10$

(c) Theoretical $g(\rho), \sigma = 100$

(d) Theoretical $g(\rho), \sigma = 1000$

Figure 5.9: Plots of theoretical $g(\rho)$. Each plot has a different value of σ; within each plot are standard error bars for different levels of k. The goal of these plots is to show how rapidly the compressed data approaches the true values as k increases. In particular we can see that the intersection with $g(\rho)$ approaches $\rho = 0$ when $\sigma = 1$.

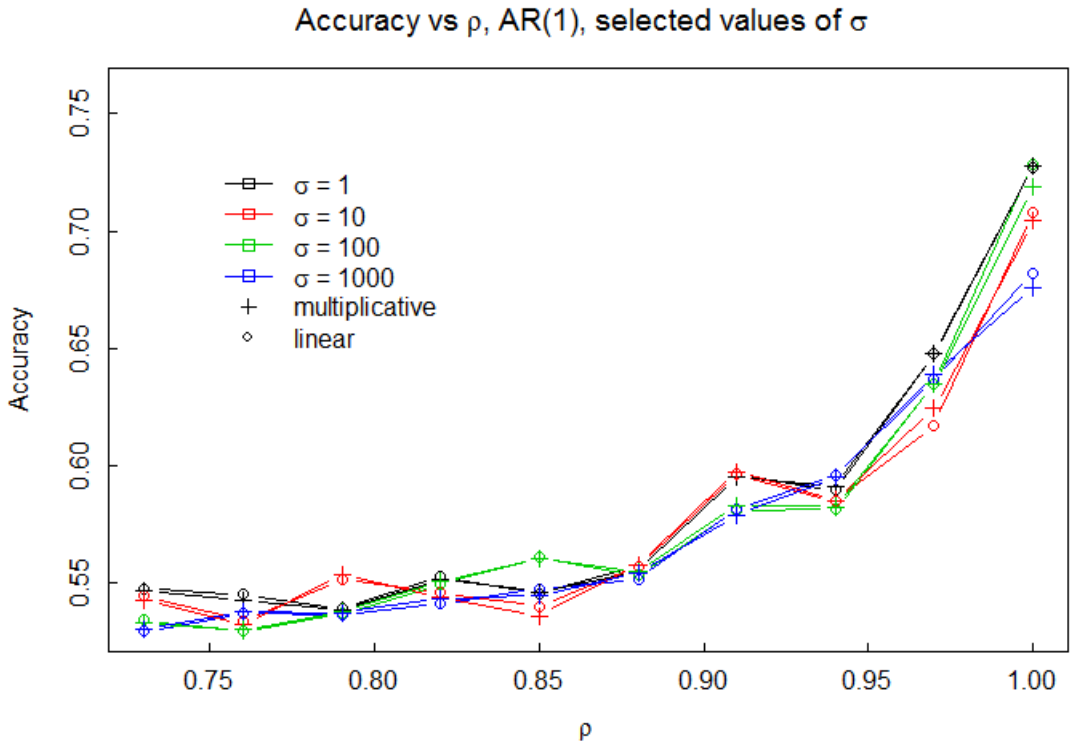


Figure 5.10: Accuracy vs ρ, regime 2 (AR(1) ρ within blocks), for various σ.

5.2.4 AR(1) simulation results

Our experiment continues with regime 2, an Autoregressive (1) ("AR(1)") model in which the within-block observations x_i and x_j have correlation $\rho^{|j-i|}$, where $|j-i|$ is the distance between the observations in the matrix: Corr $(x_i, x_j) = \rho^{|j-i|}$. As the above graphs show, this regime is much more difficult to cluster, but the relationship between ρ and performance is again clear. Clustering accuracy has high variance as accuracy increases past 0.5, which again likely has more to do with the fact that when clustering is near 0.5 - that is, the minimum - it is consistently near the minimum and thus low variance (5.11).

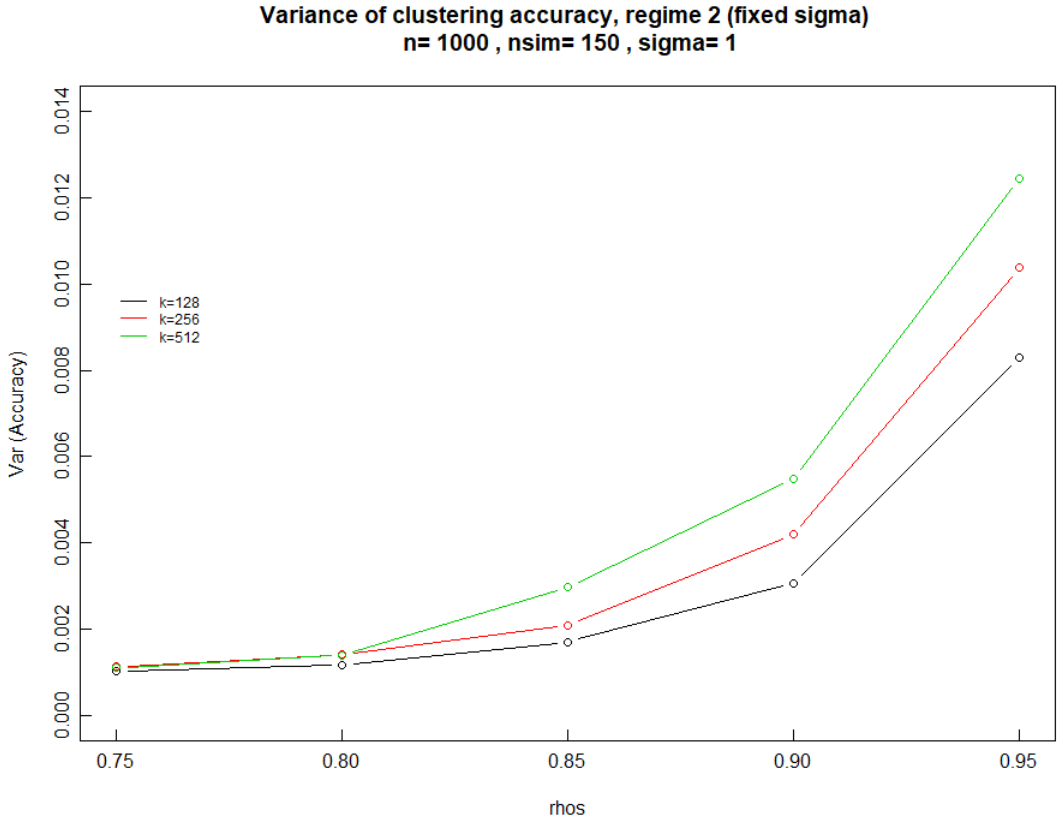


Figure 5.11: Variance of accuracy vs ρ, regime 2 (AR(1) ρ within blocks), for selected values of k.

5.3 Real data experiments

5.3.1 MNIST dataset

First, we revisited the MNIST dataset of handwritten digits. Here we present compiled plots with all digits included in each plot: each data point represents the accuracy (averaged over each simulation) of a particular pairing of digits (e.g., 0 and 1, 0 and 2, 1 and 3, and so forth up to 8 and 9). The y-axis is clustering (classification) accuracy, while the x-axis is the mean difference in correlation. Specifically, the x-axis value for Cluster C_1 and Cluster C_2, which we call $\rho_{C1,C2}$:

$$\rho_{C1,C2} = \sum_{i \in C_1} \sum_{j \in C_1} \langle x_i, x_j \rangle + \sum_{i \in C_2} \sum_{j \in C_2} \langle x_i, x_j \rangle - 2 \sum_{i \in C_1} \sum_{j \in C_2} \langle x_i, x_j \rangle \tag{5.4}$$

The above set of nine plots shows different levels of k. Across all the plots we can see that there is a clear positive relationship between $\rho_{C1,C2}$ and clustering accuracy, as well as a positive relationship between reduced dimension k and clustering accuracy. This experiment allowed us to observe very low values of σ, second set of eight plots varies σ. Performance for very low values of σ is much worse, peaking at around 70% correct, before stabilizing at around $\sigma = 0.4$ where performance approaches 100%. (Literature and our experiments suggest $\sigma = 1$ is a reasonable default.)

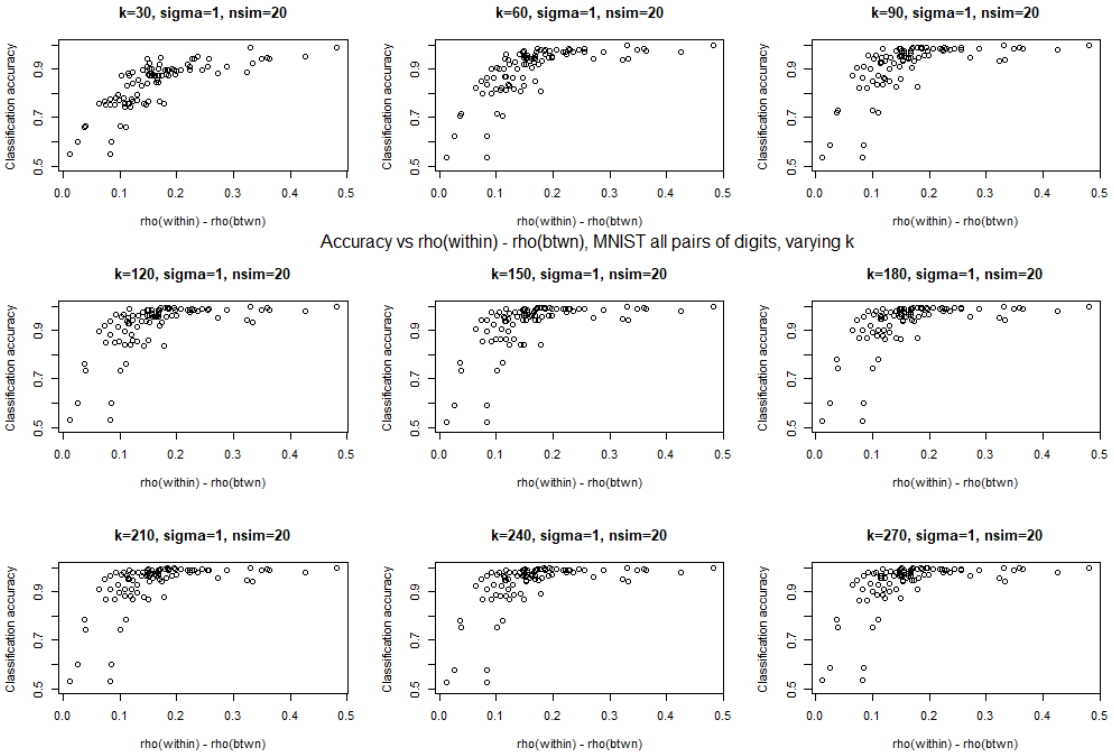


Figure 5.12: MNIST Accuracy vs $\rho_{within} - \rho_{between}$, varying k. 20 replicates, $\sigma = 1.0$.

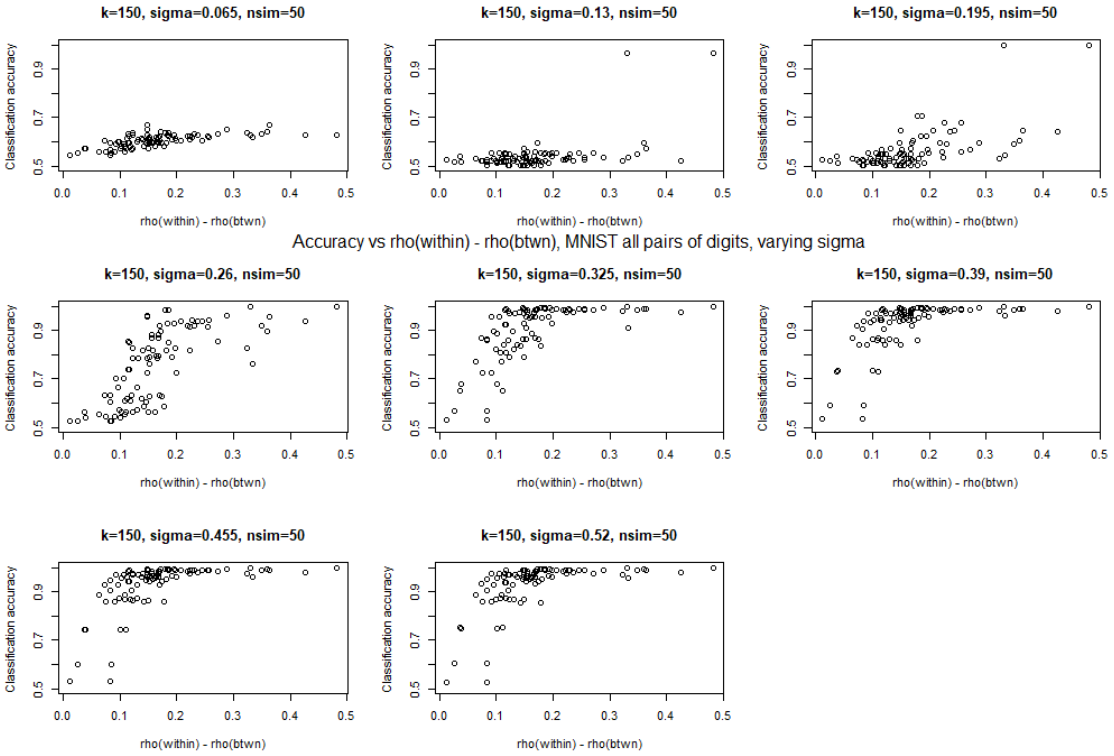


Figure 5.13: MNIST Accuracy vs $\rho_{within} - \rho_{between}$, varying σ. 50 replicates, $k = 150$.

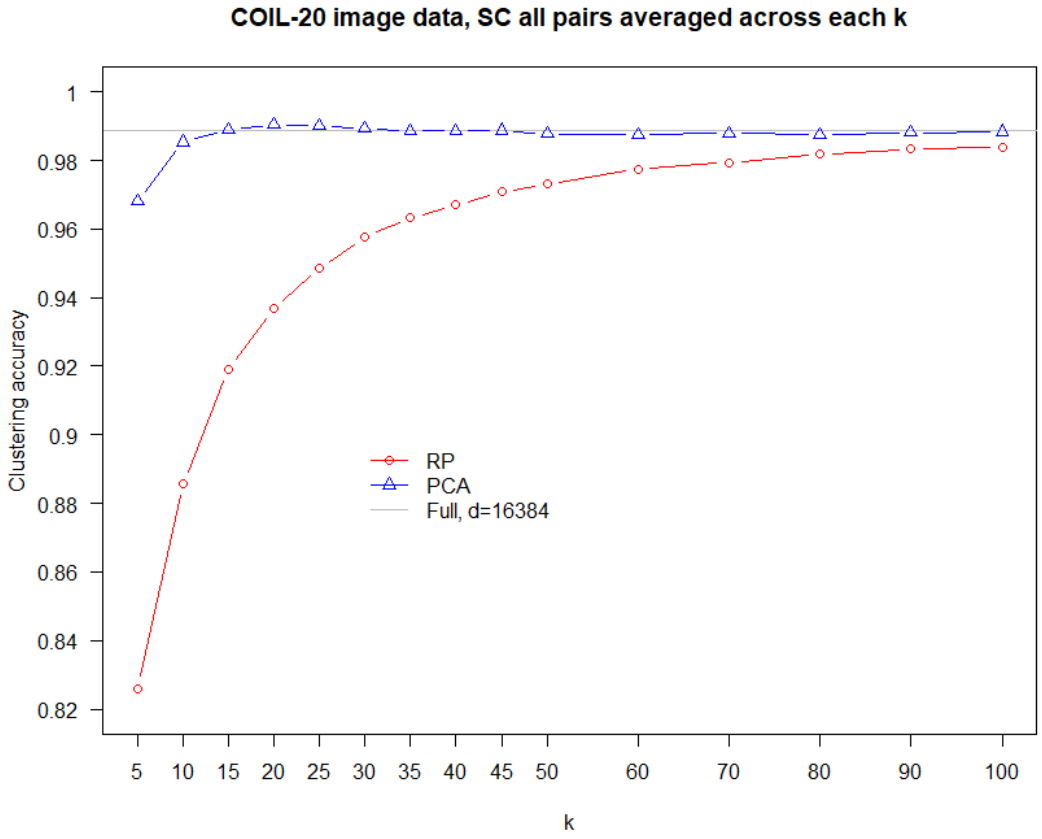


Figure 5.14: Accuracy vs k, Spectral clustering. RP vs PCA vs full data, no quantization.

5.3.2 Columbia University Image Library (COIL-20) dataset

Another dataset we looked at was the Columbia University Image Library (COIL-20) dataset of rotated images [50]. This image dataset consists of 72 angles of each of 20 objects, for 1440 total images. We first attempted all pairs spectral clustering, wherein we took each of $\binom{20}{2} = 190$ pairs of objects and clustered them all. Results are similar to our MNIST experiments. Formal time trials were not run, but it was noted that RP ran significantly faster than PCA due to the relatively large size of the images ($128 \times 128, d = 16384$).

5.4 Discussion

This dissertation began with a theoretical exploration of quantization, random projection, and spectral clustering, and has concluded with experiments that successfully employ all of these methods. We explored specific results with spectral clustering using two synthetic datasets, then performed some experiments on real data. Results show that clustering performance on a dataset compressed by RP is comparable to that of PCA, and full data for that matter, although RP is somewhat dependent on a reasonably large reduced dimension k. Quantized experiment results show that we retain almost all relevant information with quantization for $b \geq 3$. Besides the specific results we have proven, we have provided a framework for methods to apply to other machine learning algorithms or quantization methods.

There are several ways that both the theoretical and practical aspects of our work could be extended. We could compare other methods of compression besides RP and PCA. We could explore other methods of quantization besides the deterministic and stochastic methods we used. Focusing only on experiments, we could briefly add other clustering algorithms to our comparisons. With some optimization of our code, we might also attempt timed runs to compare practical run time of PCA to RP on datasets of varying size and complexity. An analysis of alternative compression methods such as structured random projection might be another feasible extension.

Appendix A: Additional proofs

In this appendix we include proofs that did not warrant inclusion in the main text. Many of these are expanded versions of proofs that are outlined in the main text; some are fairly elementary calculations, or previously established proofs that may help the reader follow along.

A.1 Section 2.1, Johnson-Lindenstrauss for inner products

Proposition A.1.1. $\langle x_i, x_j\rangle - \epsilon \cdot \|x_i\|\|x_j\| \leq \langle z_i, z_j\rangle \leq \langle x_i, x_j\rangle + \epsilon \cdot \|x_i\|\|x_j\|$

Proof. We work under our assumption of unit vector norms.

$$\begin{aligned}
\langle z_i, z_j\rangle &= \frac{1}{4}\Big(\|z_i\|^2 + \|z_j\|^2 + 2\langle z_i, z_j\rangle - \big(\|z_i\|^2 + \|z_j\|^2 - 2\langle z_i, z_j\rangle\big)\Big) \\
&= \frac{1}{4}\Big(\|z_i + z_j\|^2 - \|z_i - z_j\|^2\Big) \\
&\geq \frac{1}{4}\Big((1-\epsilon)\|x_i + x_j\|^2 - (1+\epsilon)\|x_i - x_j\|^2\Big) \\
&= \langle x_i, x_j\rangle - \frac{1}{2}\epsilon(\|x_i\|^2 + \|x_i\|^2) \\
&= \langle x_i, x_j\rangle - \epsilon \\
&\geq \langle x_i, x_j\rangle - \epsilon\langle x_i, x_j\rangle \text{ , since } \langle x_i, x_j\rangle \leq 1 \\
&= (1-\epsilon)\langle x_i, x_j\rangle
\end{aligned}$$

□

A.2 Section 2.2.2 Alternative Proof

Proposition A.2.1. *$d^2(z, z')$ is a consistent estimator for $\|x_i\|^2 + \|x_i'\|^2 - 2\langle x, x'\rangle = d^2(x, x')$.*

Proof. For this proof we note that $d^2(z, z') = \|z\|^2 + \|z'\|^2 - 2\langle z, z'\rangle$, and then we show each of $\|z\|, \|z'\|$, and $\langle z, z'\rangle$ are consistent estimators of $\|x\|, \|x'\|$, and $\langle x, x'\rangle$ respectively. We have already established that each of these is unbiased, and thus it remains to show that the variance of each converges to 0 as $k \to \infty$.

First we establish $Var\|z\|$.

$$\begin{aligned}
Var\|z\|^2 &= Var \sum_{i=1}^{k} z_i^2 \\
&= Var \sum_{i=1}^{k} (\sum_{j=1}^{d} x_j r_{ij})^2 / k \\
&= \frac{1}{k^2} Var \sum_{i=1}^{k} (x_1 r_{i1} + x_2 r_{i2} + \cdots + x_d r_{id})^2 \\
&= \frac{1}{k^2} \sum_{i=1}^{k} \left[\sum_{j=1}^{d} Var(x_j^2 r_{j1}^2) + \sum_{l=1}^{d} \sum_{m \neq l}^{d} Cov(x_l r_{jl}, x_m r_{jm}) \right] \\
&= \frac{1}{k^2} \sum_{i=1}^{k} \left[\sum_{j=1}^{d} Var(x_j^2 r_{ij}^2) + 0 \right]
\end{aligned}$$

Now, for all i, j

$$\begin{aligned}
Var(x_j^2 r_{ij}^2) &= E(x_j^4 r_{ij}^4) - (E(x_j^2 r_{ij}^2))^2 \\
&= 3x_j^4 - x_j^4 \\
&= 2x_j^4
\end{aligned}$$

Thus, going back to the original equation,

$$Var\|z\|^2 = \frac{1}{k^2}\sum_{i=1}^{k}\left[\sum_{j=1}^{d} Var(x_j^2 r_{ij}^2)\right]$$

$$= \frac{1}{k^2}\sum_{i=1}^{k}\left[\sum_{j=1}^{d} 2x_j^4\right]$$

$$= \frac{1}{k}\left[\sum_{j=1}^{d} 2x_j^4\right]$$

And so as $k \to \infty, Var\|z\| \to 0$. Now we establish $Var(\langle z, z' \rangle)$.

$$Var(\langle z, z' \rangle) = Var\left[\sum_{i=1}^{k} z_i z_i'\right]$$

$$= \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k} Cov(z_i z_i', z_j z_j')$$

$$= \frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k} E(z_i z_i' z_j z_j') - E(z_i z_i')E(z_j z_j')$$

Recall that the r_{ij} are i.i.d. $\sim N(0,1)$, and so we have the following moments:

$E[r_{ij}] = 0, E[r_{ij}^2] = 1, E[r_{ij}^2 r_{il}^2] = 1, E[r_{ij}^3 r_{il}] = 0, E[r_{ij}^4] = 3$

Now, for all i, and noting that the $r_{ij} \sim N(0,1)$ and are independent from each other,

$$\begin{aligned}
E(z_i z_i') &= E\left[\sum_{l=1}^{d} x_l r_{il} \sum_{m=1}^{d} x_m' r_{im}\right] \\
&= E\left[(x_1 r_{i1} + \cdots + x_d r_{id})(x_1' r_{j1} + \cdots + x_d' r_{jd})\right] \\
&= E\left[\sum_{l=1}^{d} x_l x_l' r_{il}^2\right] + E\left[\sum_{l=1}^{d} \sum_{m \neq l} x_l r_{il} x_m' r_{im}\right] \\
&= \left[\sum_{l=1}^{d} x_l x_l' E[r_{il}^2]\right] + \sum_{l=1}^{d} \sum_{m \neq l} x_l x_m' E[r_{il}] E[x_{im}] \\
&= \langle x, x' \rangle + 0
\end{aligned}$$

Now we solve for $E(z_i z_i' z_j z_j')$. To do so we break into two cases, $i = j$ (for which there are k cases) and $i \neq j$ (for which there are $k(k-1)$ cases). For the case that $i \neq j$, we note that $(z_i, z_j, z_i', z_j)'$ form a multivariate normal random vector, and so by Isserlis' theorem (details Appendix A) and equation A.2 above

$$\begin{aligned}
E(z_i z_i' z_j z_j') &= E(z_i z_i') E(z_j z_j') + E(z_i z_j) E(z_i' z_j') + E(z_i' z_j) E(z_i z_j') \\
&= \langle x, x' \rangle^2 + 0 + 0
\end{aligned}$$

Thus, putting it together, we have

$$
\begin{aligned}
E(z_i z_i' z_i z_i') &= E(z_i^2 z_i'^2) \\
&= E\left[(x_1^2 r_{i1}^2 + x_2^2 r_{i2}^2 + \cdots + x_d^2 r_{id}^2)(x_1'^2 r_{i1}^2 + x_2'^2 r_{i2}^2 + \cdots + x_d'^2 r_{id}^2)\right] \\
&= E(x_1^2 x_1'^2 r_{i1}^4 + \cdots + x_1^2 x_1'^2 r_{i1}^4) + E\left[\sum_{l=1}^{d}\sum_{m\neq l} x_l^2 x_m'^2 r_{il}^2 r_{im}^2\right] \\
&= 3\sum_{l=1}^{d} x_l^2 x_l'^2 + \sum_{l=1}^{d}\sum_{m\neq l} x_l^2 x_m'^2 \\
&= 3\langle x, x'\rangle^2 + \|x\|_2^2 \cdot \|x'\|_2^2 - \langle x, x'\rangle^2
\end{aligned}
$$

Putting these parts together, we have:

$$
\begin{aligned}
&\frac{1}{k^2}\sum_{i=1}^{k}\sum_{j=1}^{k} E(z_i z_i' z_j z_j') - E(z_i z_i')E(z_j z_j') \\
&= \frac{1}{k^2}\left(k\left[3\langle x, x'\rangle^2 + \|x\|_2^2 \cdot \|x'\|_2^2 - \langle x, x'\rangle^2\right] + (k^2 - k)\langle x, x'\rangle^2 - k^2\langle x, x'\rangle^2\right) \\
&= \frac{1}{k}\left(\|x\|_2^2 \cdot \|x'\|_2^2 + \langle x, x'\rangle^2\right)
\end{aligned}
$$

Thus, as $k \to \infty$, $Var\langle z, z'\rangle \to 0$, and (recalling that $E\langle z, z'\rangle = \langle x, x'\rangle$), $\langle z, z'\rangle$ is a consistent estimator for $\langle z, z'\rangle$.

In all, we have that $\|z\|, \|z'\|$, and $\langle z, z'\rangle$ are consistent estimators of $\|x\|, \|x'\|$, and $\langle x, x'\rangle$ respectively. Since $d^2(z, z') = \|z\|^2 + \|z'\|^2 - 2\langle z, z'\rangle$, $d^2(z, z')$ is a consistent estimator for $\|x_i\|^2 + \|x_i'\|^2 - 2\langle x, x'\rangle = d^2(x, x')$. □

A.3 Section 3.2 Proof using Isserlis' theorem

Proposition A.3.1. *For bivariate normal vectors Z, Z', $E(Z^2 Z'^2) = Var(Z)\,Var(Z') + 2Cov(Z, Z')^2 = 1 + 2\rho^2$*

Proof. We begin by stating Isserlis' Theorem, then apply to our case.

Isserlis' Theorem

For a multivariate, 0-mean normal vector $(X_1 X_2 \dots X_{2n})$ with $E(X_i) = 0$ for all $i \in \{1, ..., 2n\}$,

$$E(X_1 X_2 ... X_{2n}) = \sum \prod E(X_i X_j), \text{ and in particular}$$

$$E(X_1 X_2 X_3 X_4) = E(X_1 X_2)E(X_3 X_4) + E(X_1 X_3)E(X_2 X_4) + E(X_1 X_4)E(X_2 X_3)$$

Where the sum is over all disjoint pairs of the $\{X_i, X_j\}$, and the product is over all distinct pairings within of said random variables. [51]

Back to $E(z_i z_i' z_j z_j')$, we now solve for the case that $i = j$, we have

$$\begin{aligned}
E(z_i z_i' z_i z_i') &= E(z_i^2 z_i'^2) \\
&= E\left[(x_1^2 r_{i1}^2 + x_2^2 r_{i2}^2 + \cdots + x_d^2 r_{id}^2)(x_1'^2 r_{i1}^2 + x_2'^2 r_{i2}^2 + \cdots + x_d'^2 r_{id}^2)\right] \\
&= E(x_1^2 x_1'^2 r_{i1}^4 + \cdots + x_1^2 x_1'^2 r_{i1}^4) + E\left[\sum_{l=1}^{d} \sum_{m \neq l} x_l^2 x_m'^2 r_{il}^2 r_{im}^2\right] \\
&= 3\sum_{l=1}^{d} x_l^2 x_l'^2 + \sum_{l=1}^{d} \sum_{m \neq l} x_l^2 x_m'^2 \\
&= 3\langle x, x' \rangle^2 + \|x\|_2^2 \cdot \|x'\|_2^2 - \langle x, x' \rangle^2
\end{aligned}$$

□

A.4 Section 3.2 Lemmas

Lemma A.4.1. *Bias*$(\hat{\rho}) = 0$

Proof. We use conditioning: $E(QQ') = E[E(QQ'|Z, Z')]$, treating Z, Z' as constants. Note that this proof applies to any number of bits.

$$E\left[QQ'|Z, Z'\right] = b^2 P(Q = b, Q' = b) + ab \cdot P(Q = a, Q' = b) + ba \cdot P(Q = b, Q' = a)$$

Now, assuming $a = -b$:

$$\begin{aligned} E\left[QQ'|Z, Z'\right] &= \frac{1}{4b^2}[b^2(z+b)(z'+b) + b^2(b-z)(b-z') - b^2(z+b)(b-z') \\ &\quad + b^2(z'+b)(b-z)] \\ &= z'(b+z-b+z)/2 \\ &= zz' \end{aligned}$$

This gives us:

$$\begin{aligned} E\left[E\left[QQ'|Z, Z'\right]\right] &= E(zz') \\ &= Cov(ZZ') - E(Z)E(Z') = \rho \end{aligned}$$

□

Lemma A.4.2. *Var*$[E\left[QQ'|Z, Z'\right]] = 1 + \rho^2$

Proof.

$$\begin{aligned} Var\left[E\left[QQ'|Z,Z'\right]\right] &= Var[zz'] \\ &= E(z^2z'^2] - [E(zz')]^2 \\ &= 1 + 2\rho^2 - \rho^2 \\ &= 1 + \rho^2 \end{aligned}$$

□

A.5 Delta Method using Taylor Expansion Detailed

Our application of the delta method has its basis with the Taylor expansion / approximation of a function $f(\cdot)$ at a point, using the Lagrange form of the remainder for the second term:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x + p \cdot h), p \in (0,1)$$

In the context of an estimator $\hat{\rho}$ of ρ, we take a Taylor expansion around E$[\hat{\rho}]$, i.e. in the above formula we set $x = E(\hat{\rho}), h = \hat{\rho} - E(\hat{\rho})$, and take $\tilde{\rho} = max(\rho, \hat{\rho})$:

$$f(\hat{\rho}) = f(E(\hat{\rho})) + f'(E(\hat{\rho}))(\hat{\rho} - E(\hat{\rho})) + \frac{(\hat{\rho} - E[\hat{\rho}])^2}{2!}f''(E(\tilde{\rho}))$$

So taking expectation or variance of both sides we have:

$$\begin{aligned} E\left[f(\hat{\rho})\right] &= E\left[f(E(\hat{\rho})) + f'(E(\hat{\rho}))(\hat{\rho} - E(\hat{\rho})) + \frac{(\hat{\rho} - E[\hat{\rho}])^2}{2!}f''(E(\tilde{\rho}))\right] \\ &= f(E[\rho]) + 0 + \frac{E[(\hat{\rho} - E[\hat{\rho}])^2]}{2}f''(E(\tilde{\rho}))] \end{aligned}$$

$$Var\left[f(\hat{\rho})\right] = Var\left[f(E(\hat{\rho})) + f'(E(\hat{\rho}))(\hat{\rho} - E(\hat{\rho})) + \frac{(\hat{\rho} - E[\hat{\rho}])^2}{2!} f''(E(\tilde{\rho}))\right]$$

$$= 0 + Var(\hat{\rho}) \cdot \left[f'(\hat{\rho})^2\right] + R$$

We will often use the Lagrange form of the remainder after the first term, $R = \frac{1}{2} f''(E(\tilde{\rho}))(\hat{\rho} - E(\hat{\rho}))^2$, and

$$Var(R) = Var\left(\frac{1}{2} f''(E(\tilde{\rho}))(\hat{\rho} - E(\hat{\rho}))^2\right)$$

$$= \frac{1}{4} f''(E[\tilde{\rho}])^2 Var\left((\hat{\rho} - E[\hat{\rho}])^2\right)$$

A.6 Section 3.2.3 discssion on selection of b

The estimator $(\hat{\rho_{SBQ}})$ is unbiased only if $b \geq \max_{i \in [1,k]} z_i$. It is feasible to go through the data post-projection to set b equal to the maximum over observations $(z_i j)$, but this would be an inefficient process is of order $O(nk)$. More practically, we can choose b depending of n, such as proportional to $\sqrt{\log nk}$, which is the order of $E[\max_{1 \leq i \leq n} |Z_i|])$. With this choice of b the bias becomes negligible.

In theory, for a given sample, b should be set to be the maximum of the sample so that Z is never greater than b, i.e. $P(Z > b) = 0$. This operation of determining the min and max of each Z would be an operation of $\mathrm{O}\left(\frac{1}{k}\right)$, and perhaps not feasible for practical application. One option for choosing b could instead be to set b such that the probability of a projected value $z_l > b$, given a sample of n observations and reduced dimension k, is $\alpha = 0.001$ (say):

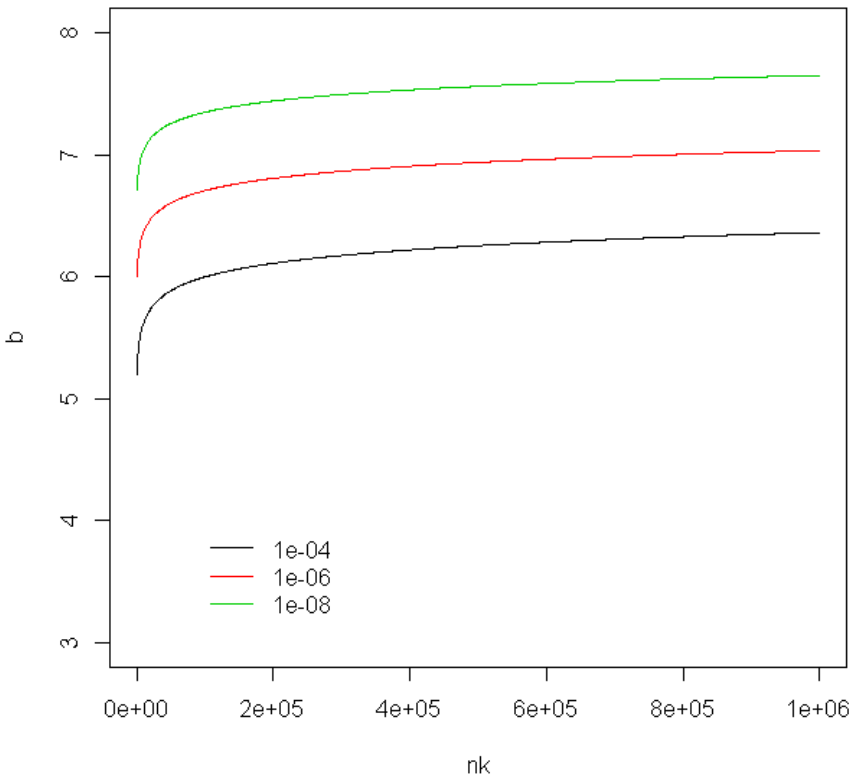


Figure A.1: Values of b corresponding to various α and $n \cdot k$. We can see that even for extreme values, setting $b = 7$ is sufficiently high.

$$P(z_{(n)} > b) = \alpha \implies P(z_1, ..., z_{nk} < b) = 1 - \alpha$$

$$P(z_1, ..., z_{nk} < b) = (\Phi(b))^{nk} = 1 - \alpha$$

$$\Phi(b) = (1 - \alpha)^{1/nk}$$

$$b = \Phi^{-1}\left((1 - \alpha)^{1/nk}\right)$$

We could instead choose b according to n, for example $b = O\left(\sqrt{\log(k)}\right)$, which is of the

order of $E[\max_{1\leq i\leq k}|z_l|])$. Another option might be to follow (Royston 1982, Blom 1958, proper citations coming if we use this), wherein we set

$$b \propto E[z_{(k)}] \approx \Phi\frac{(i-\nu)}{k-2\alpha+1}$$

where $\Phi(\cdot)$ is the cdf of a standard normal random variable and $\nu \approx 0.375$.

A.7 Section 3.2.4

We substitute the below in to the work in section 3.2.4, in the calculation of $E(E(Q^2Q'^2|Z,Z'))$.

$$
\begin{aligned}
&E(E(Q^2Q'^2|Z,Z')) \\
&= E\Big[E(Q^2|Z)\cdot E(Q'^2|Z')\Big] \\
&= E\Big[(2b^2(I(Z>b))+a^2I(Z\in(-a,a])+(Z(b+a)-ab)I(Z\in(a,b) \\
&\qquad -(Z(a+b)+ab)(I(Z\in(-b,-a])) \\
&\qquad \cdot 2b^2(I(Z>b))+a^2I(Z\in(-a,a])+(Z(b+a)-ab)I(Z\in(a,b) \\
&\qquad -(Z(a+b)+ab)(I(Z\in(-b,-a])\Big] \\
&= 4b^2P(Z>b,Z'>b)+a^4P(Z,Z\in(-a,a]) \\
&\qquad +E[ZZ'(a+b)^2+(Z+Z')(a+b)ab+a^2b^2|Z,Z'\in(a,b]]\cdot P(Z,Z'\in(a,b]) \\
&\qquad +E[ZZ'(a+b)^2-(Z+Z')(a+b)ab+a^2b^2|Z,Z'\in(-b,-a]]\cdot P(Z,Z'\in(-b,-a]) \\
&\qquad +P(Z\in(-a,a],Z\in(a,b])a^2 \\
&\qquad \cdot[((a+b)E(Z'|Z'\in(a,b))+ab)+(a+b)E(Z|Z\in(a,b])+ab) \\
&\qquad -((a+b)E(Z'|Z'\in(-b,-a])-ab)-((a+b)E(Z|Z\in(-b,-a])-ab))] \\
&= a^4P(Z,Z\in(-a,a])+4a^2(E[Z\cdot I_{Z\in(-a,a),Z'>a}(a+b)]-ab\cdot P(Z\in(-a,a),Z'>a)) \\
&\qquad +2E[ZZ'I_{Z>a,Z'>a}(a+b)^2-(Z+Z')(a+b)ab\cdot I_{Z>a,Z'>a}] \\
&\qquad +2a^2b^2\cdot P(Z>a,Z'>a) \\
&\qquad -2E[ZZ'I_{Z>a,Z'<-a}(a+b)^2+(a+b)ab(Z-Z')\cdot I_{Z>a,Z'<-a}-a^2b^2I_{Z>a,Z'<-a}]
\end{aligned}
$$

Lemma A.7.1. *$Var[E\left[QQ'|Z,Z'\right]] = 1+\rho^2$*

Proof.

$$\begin{aligned} Var\left[E\left[QQ'|Z,Z'\right]\right] &= Var[zz'] \\ &= E(z^2z'^2] - [E(zz')]^2 \\ &= 1+2\rho^2-\rho^2 \\ &= 1+\rho^2 \end{aligned}$$

□

A.7.1 Section 4.1, full calculation using Taylor expansions for $E(\hat{\rho})$

$$E\left[\hat{\rho}\right] = E\left[\frac{\langle z, z' \rangle}{\|z\|\|z'\|}\right] = E[\frac{a}{\sqrt{bc}}]$$

$$= E\left[f\left(E\begin{pmatrix} a \\ b \\ c \end{pmatrix}\right) + \frac{1}{2}\begin{pmatrix} a-\rho & b-1 & c-1 \end{pmatrix} \nabla^2 f(\tilde{a},\tilde{b},\tilde{c}) \begin{pmatrix} a-\rho \\ b-1 \\ c-1 \end{pmatrix}\right]$$

$$\leq \rho +$$

$$E\left[\begin{pmatrix} a-\rho & b-1 & c-1 \end{pmatrix} \begin{pmatrix} 0 & -\frac{1}{2}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{1}{2}} & -\frac{1}{2}\tilde{b}^{-\frac{1}{2}}\tilde{c}^{-\frac{3}{2}} \\ -\frac{1}{2}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{1}{2}} & \frac{3}{4}\tilde{a}\tilde{b}^{-\frac{5}{2}}\tilde{c}^{-\frac{1}{2}} & \frac{1}{4}\tilde{a}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{3}{2}} \\ -\frac{1}{2}\tilde{b}^{-\frac{1}{2}}\tilde{c}^{-\frac{3}{2}} & \frac{1}{4}\tilde{a}\tilde{b}^{-\frac{3}{2}}\tilde{c}^{-\frac{3}{2}} & \frac{3}{4}\tilde{a}\tilde{b}^{-\frac{1}{2}}\tilde{c}^{-\frac{5}{2}} \end{pmatrix} \begin{pmatrix} a-\rho \\ b-1 \\ c-1 \end{pmatrix}\right]$$

$$= \rho + \frac{1}{2}E\Big[(a-\rho)\left(-\frac{b+c+2}{2}\right) + (b-1)\left(-\frac{(a-\rho)}{2} + \frac{3\tilde{a}(b-1)}{4} + \frac{\tilde{a}(c-1)}{4}\right) +$$

$$+ (c-1)\left(-\frac{(a-\rho)}{2} + \frac{\tilde{a}(b-1)}{4} + \frac{3\tilde{a}(c-1)}{4}\right)\Big]$$

$$= \rho + E\Big[(a-\rho)(2-b-c) + \frac{3\tilde{a}[(b-1)^2 + (c-1)^2] + 2\tilde{a}(b-1)(c-1)}{8}\Big]$$

$$
\begin{aligned}
E\left[\hat{\rho}\right] &= \rho - \rho + E(a) \\
&\quad - \frac{E(ab) - E(ac) + E(\rho b) + E(\rho c)}{2} + \frac{3\tilde{a}(V(b) + V(c)) + 2\tilde{a}E[(b-1)(c-1)]}{8} \\
&= \rho + 2\rho - (\rho + \frac{2\rho}{k}) - 2\rho + \rho + \rho + \frac{3\tilde{a}(2 \cdot \frac{2}{k} + 2\tilde{a}(1 + \frac{\rho^2}{k} - 1 - 1 + 1)}{8} \\
&= \rho + \frac{2\rho}{k} + \frac{3\tilde{a}}{k} + \frac{\tilde{a}\rho^2}{2k} \\
&= \rho + \frac{2\rho + \tilde{a}(3 + \rho^2/2)}{k}
\end{aligned}
$$

And thus $\text{Bias}(\hat{\rho}) = O\left(\frac{1}{k}\right)$.

Appendix B: Spectral Clustering Experiment Details and Results

B.1 Spectral Clustering Experiment Details

This appendix describes the spectral clustering experiment in detail. Spectral Clustering is a clustering algorithm that uses the eigenvalues of the data's graph similarity matrix (described below) [52] [1]. It is a relaxation of the graph cut problem. One of its advantages is that it relies only on pairwise distances; since this is what random projection preserves, we would expect spectral clustering to be a good application of RP.

Many of our experiments involved comparing RP to PCA. The algorithms are similar on their faces: both take existing data and reduce to lower dimension, and can be represented as a matrix multiplication. Whereas RP gives a projection of the data on a random subspace, PCA gives a projection on a sub-space that provides the best linear approximation of the data. Of course, this is done at the expense of computational complexity. Each principal component minimizes the sum of squared distance between all points and their projection in the subspace.

PCA maximizes the amount of variance accounted for with a given number of dimensions. In theory, we expect PCA to give better accuracy at the cost of processing time for a given k; for lower k, likely much better, as PCA may provide noise reduction.

For our experiment, we combine spectral clustering with each of the two methods of dimensionality reduction, random projection and principal component analysis.

The spectral clustering algorithm we used was introduced in [46]. We followed the implementation outlined in [1] with minimal modification to incorporate our dimensionality

reduction. We present the entire algorithm below, clustering data X to κ clusters.

B.1.1 Spectral Clustering Implementation

(1) Project data using RP (or PCA for comparison) to reduce dimensionality, ie project $X, x_i \in \mathbb{R}^d$ to obtain $Z = RX, z_i \in \mathbb{R}^k$.

(1a) **PCA implementation**

i. Find singular value decomposition of X, ie U, Σ, V such that $X = U\Sigma V^T$, with $X, U \in \mathbb{R}^{n\times d}, \Sigma \in \mathbb{R}^{d\times d}, V \in \mathbb{R}^{d\times d}$

ii. Create $R \in \mathbb{R}^{d\times k}$ by taking first k vectors of V as columns, ie $R_{d\times k} = v_1, ..., v_k,$ with $v_i \in \mathbb{R}^{d\times 1}$.

iii. Create reduced data matrix $Z = RX$.

(1b) **RP implementation**

i. Create $R \in \mathbb{R}^{d\times k}$ where $r_{ij} \sim N(0, 1)$.

ii. Create reduced data matrix $Z = RX$.

(2) Create quantized data matrix $Q = Q_u(Z)$, where $Q_u(\cdot)$ represents some general element-wise quantization of Z. This can be done in one of several ways:

(2a) Deterministic Quantization, described in section 3.1.1.

(2b) Stochastic Quantization, described in section 3.1.2.

(2c) No quantization at all, i.e. $Q = Z$.

(3) Construct graph and associated Laplacian matrices.

(3a) Calculate similarities between all pairs of data points in $\mathbb{R}^k$. For our experiment, we used the Gaussian similarity, ie $s_{ij} = s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$.

Aside: For data that has been compressed to quantized space $(\mathcal{M}^{\pm})^k$, we could calculate distances between points in $(\mathcal{M}^{\pm})^k$ via alternative methods such as hamming distance $d_h(x,y) = \frac{1}{k}\sum_{j=1}^{k} I_{(x_j \neq y_j)}$; i.e. for all $i,j, d(q_i,q_j) \in \{0, \frac{1}{k}, \frac{2}{k}, ..., 1\}$.

Another, simpler, option involves simply substituting the coded value (typically midpoint) or even cut points in to other distance measures and pretending that the data are from the real line. Alternatively, we can use inner product (or cosine similarity), $d(x,y) = \langle x, y \rangle$. If data are normalized then $d(x,y) \in [0,1]$ and the same weight formula below can be applied. In this case the weights are in [0,1].

(3b) Construct the K-nearest neighbors graph W using these similarities. To ensure symmetry, we connect an edge between x_i and x_j if x_i is amongst the K nearest neighbors of y_i, or if y_i is amongst the K nearest neighbors of x_i. The K^{th} nearest neighbor of x_i is defined as the data point with the K^{th} highest similarity. In either case, $w_{ij} = s_{ij}$ is then stored in weight matrix W. Otherwise the i,j entry is 0. Note $w_{ii} = 1$.

We would like to emphasize here that capital K, the number of nearest neighbors used to generate W, has nothing to do with k, the reduced dimension, or κ, the number of clusters.

If using normalized distances, the weight of each edge $w(x_i, x_j)$ can be set to $1 - d(x_i, x_j) \geq 0$, since the algorithm relies on similarities instead of distances.

(3c) Generate degree matrix D, diagonal with entries $d_{ii} = \sum_j w_{ij}, i = 1, \ldots, n$.

(3d) Generate Laplacian matrices. First, create unnormalized Laplacian $L = D - W$. Then create normalized, symmetric Laplacian $L_{sym} = D^{-1/2}LD^{-1/2}$

Note that L is positive semi-definite, symmetric, and has n real eigenvalues ≥ 0 (including 0, which has eigenvector $\mathbf{1}$, the vector composed of all 1's). L_{sym} has eigenvalue 0 with eigenvector $D^{1/2}1$, and is positive semi-definite, with n non-negative real eigenvalues.

(4) Compute first κ eigenvectors of L, $v_1, ..., v_\kappa$, where "first" is defined as those associated with smallest κ eigenvalues (recalling that L is positive semi-definite).

(4a) Create matrix $U \in \mathbb{R}^{n \times \kappa}$ by arranging $v_1, ..., v_\kappa$ into columns.

(4b) Form matrix $T \in \mathbb{R}^{n \times \kappa}$ by normalizing U by rows. Let y_i be the $\kappa \times 1$ vector corresponding to the i^{th} row of U.

(5) Cluster the points y_i in $\mathbb{R}^\kappa$ to κ clusters C_i. This may be done with k-means or some other method. The clustering of the data x_i follows this clustering.

B.1.2 Experiment 1 Details

Our first experiment involved running the above algorithm, comparing PCA and RP separately. The steps were repeated for k from 10 to 300, in increments of 10, for both PCA and RP. Since RP had yet to converge to full data performance, we then continued running k up to 4000 in increments of 200.

For each iteration of the simulation (i.e., for each value of k), 20 RPs were calculated and SCs performed for each projection. Since PCA is deterministic it was only run once for each k. The R function kmeans was used for this experiment, with 5 random starts.

Our database is the Mixed National Institute of Standards and Technology database (MNIST, [47]), a database of handwritten digits compiled by the USPS. Each handwritten digit is represented by a 28×28 grey scale bitmap, with each pixel having one of 256 shades in the grey scale. Thus, one row of the database contains 784 columns corresponding to the shade of a pixel in the bitmap.

We stripped the training database down to several of the more easily-identifiable digits for the purposes of this experiment, namely 0, 1, 2, 6, and 9. For this stage, a sample of 1000 digits were used from the original training database of about 60000 bitmaps. Thus our data set X consists of about 200 images for each of 5 digits, that is 1000 rows.

B.1.3 Experiment 1 Results

For this experiment we used the Rand index [48] to measure similarity between the correct clustering and the calculated clustering. The Rand Index calculates the fraction of pairings that agree between the two clusterings, and thus may be slightly higher or lower than actual (mis)-classification rate but is roughly equivalent. We then calculated the min, max, and average Rand indexes.

Figure B.1 plots Rand index against reduced dimension k. PCA performance surpasses full data performance around $k = 50$, peaks at $k = 90$, and descends from there until it converges with full data performance at around $k = 220$.

RP performance is consistently lower than performance of both full data and PCA.

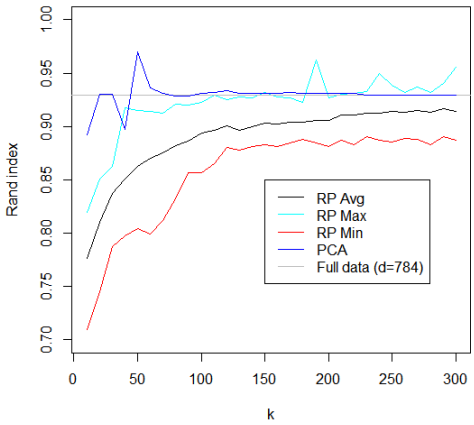


Figure B.1: RP and PCA, $k \leq 300$

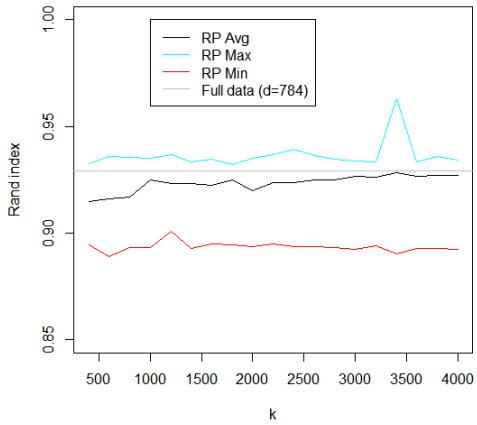


Figure B.2: RP, $k \geq 400$

Figure B.3: Experiment 1 Results

However, performance does begin to converge to that of full data, as theory dictates. The extended graph in figure $B.2$ shows this.

B.1.4 Experiment 2 Details

A second experiment was conducted, this time dividing the data into pairs of digits and clustering these. This two-cluster experiment allowed us to use thresholding, a more controllable method of clustering than kmeans, as well as ratio cut instead of Rand index to measure performance. Ratio cut is a balanced cut metric that measures the difference between clusters of a particular graph partitioning; the formula for the K-cluster case is demonstrated below. [53] [54]

$$RatioCut(C, V\backslash C) = \sum_{l=1}^{K} \frac{cut(C_l, V\backslash C_l)}{|C_l|}, \text{ where}$$

$$cut(C_l, V\backslash C_l) = \sum_{i \in C, j \in V\backslash C} w_{ij}$$

We outline the experiment here, referring to the algorithm in B.1.1 where necessary.

Experiment 2 Procedure

(1) Divide data into pairs of digits, i.e. each of $\binom{10}{2} = 45$ pairs, to attempt to cluster into 2 clusters. We call any particular data set X.

(2) Create compressed data set $Z = XR$ via either RP or PCA. Use reduced dimension k from 10 through 200 via increments of 10.

(3) Create K-nearest neighbor graph W of compressed data Z using Gaussian similarity, degree matrix D, Laplacian L as in step (2) above.

(4) Compute the eigenvectors $v_1, ..., v_n$ of L, but store only v_2 (the one corresponding to smallest non-zero eigenvalue).

(5) Threshold the eigenvector y_2:

(5a) Sort y_2 in ascending order to generate sorted vector $y_{sort} = \{y_{2,(1)}, \cdots, y_{2,(n)}\}$.

(5b) For $j \in \{1, 2, \cdots, n-1\}$, divide y_{sort} into two clusters by defining cluster $C = \{y_{2,(1)}, \cdots, y_{2,(j)}\}$. The remaining elements $y_{sort} \backslash C$ compose the other cluster $V \backslash C$. Calculate $RatioCut(C, V \backslash C)$ for each threshold j, using the weight matrix W of the compressed data.

(5c) After looping through all $n-1$ possible thresholds, choose the one with the lowest ratio cut value. Call the optimal clustering C_z.

(6) Measure results by comparing to same algorithm run on full data X, again using RatioCut or NormalizedCut as the metric.

(6a) Run above steps (1) through (5), except create eigenvector y_2 from full data X. Call this optimal clustering C_x.

(6b) Evaluate performance of data compression by comparing C_x to C_z, via RatioCut $(C_x, V \backslash C_x)$ - RatioCut $(C_z, V \backslash C_z)$. Note that at this measurement step the affinity matrix W to be used in the RatioCut calculation is that of the full data, not that of the compressed data.

Lower values indicate better performance. By using different compression methods to create Z, they can be compared to either other and the full data performance.

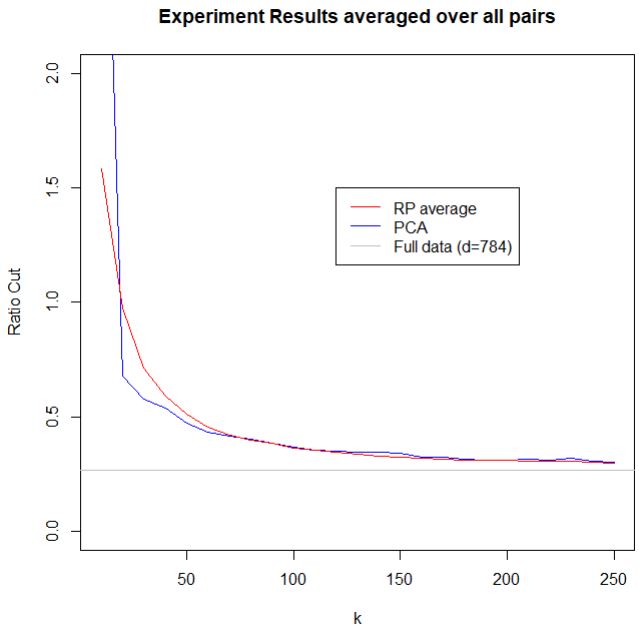


Figure B.4: Experiment 2 Results

B.1.5 Experiment 2 Results

For this experiment we measure performance of compression methods via RatioCut, as described above. Lower values of ratio cut indicate more dissimilar partitions, i.e. better performance. We present averaged results over all 45 pairs: that is, for any particular point .

Figure B.4 plots ratio cut against reduced dimension k. PCA performance climbs steadily towards full data performance, although it does not quite converge to full data.

RP performance is consistently lower than performance of PCA at the same reduced dimension k, besides at $k = 10$ where PCA performs very poorly. The difference is small however, with ratio cut values very similar to that of PCA for $k < 90$, and almost identical from around $k = 100$.

Bibliography

[1] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[2] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *Journal of Machine Learning Research*, vol. 7, no. Oct, pp. 1963–2001, 2006.

[3] H. Zare, P. Shooshtari, A. Gupta, and R. R. Brinkman, "Data reduction for spectral clustering to analyze high throughput flow cytometry data," *BMC bioinformatics*, vol. 11, no. 1, p. 403, 2010.

[4] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 287–296.

[5] M. Liu, Z. Shang, and G. Cheng, "Sharp theoretical analysis for nonparametric testing under random projection," in *Conference on Learning Theory*, 2019, pp. 2175–2209.

[6] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[7] P. Li, M. Mitzenmacher, and M. Slawski, "Quantized random projections and non-linear estimation of cosine similarity," in *Advances in Neural Information Processing Systems*, 2016, pp. 2748–2756.

[8] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2938–2945.

[9] W. Kong and W.-J. Li, "Double-bit quantization for hashing," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[10] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, 2009.

[11] M. Garey, D. Johnson, and H. Witsenhausen, "The complexity of the generalized lloyd-max problem (corresp.)," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 255–256, 1982.

[12] P. Carbone and D. Petri, "Performance of stochastic and deterministic dithered quantizers," in *IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (Cat. No. 99CH36309)*, vol. 3. IEEE, 1999, pp. 1653–1658.

[13] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2001, pp. 274–281.

[14] R. L. Dykstra, "An algorithm for restricted least squares regression," *Journal of the American Statistical Association*, vol. 78, no. 384, pp. 837–842, 1983.

[15] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.

[16] B. Schölkopf, "The kernel trick for distances," in *Advances in neural information processing systems*, 2001, pp. 301–307.

[17] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[18] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[19] P. Li, T. J. Hastie, and K. W. Church, "Improving random projections using marginal information," in *International Conference on Computational Learning Theory*. Springer, 2006, pp. 635–649.

[20] S. Dasgupta, "Learning mixtures of gaussians," in *Foundations of computer science, 1999. 40th annual symposium on*. IEEE, 1999, pp. 634–644.

[21] ——, "Experiments with random projection," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 143–151.

[22] C. Boutsidis, A. Zouzias, and P. Drineas, "Random projections for k-means clustering," in *Advances in Neural Information Processing Systems*, 2010, pp. 298–306.

[23] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 245–250.

[24] D. Fradkin and D. Madigan, "Experiments with random projections for machine learning," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 517–522.

[25] H. Xie, J. Li, Q. Zhang, and Y. Wang, "Comparison among dimensionality reduction techniques based on random projection for cancer classification," *Computational biology and chemistry*, vol. 65, pp. 165–172, 2016.

[26] S. Deegalla and H. Bostrom, "Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification," in *2006 5th International Conference on Machine Learning and Applications (ICMLA'06)*. IEEE, 2006, pp. 245–250.

[27] N. Goel, G. Bebis, and A. Nefian, "Face recognition experiments with random projection," in *Biometric Technology for Human Identification II*, vol. 5779. International Society for Optics and Photonics, 2005, pp. 426–437.

[28] A. Bouzalmat, N. Belghini, A. Zarghili, J. Kharroubi, and A. Majda, "Face recognition using neural network based fourier gabor filters & random projection," *International Journal of Computer Science and Security (IJCSS)*, vol. 5, no. 3, p. 376, 2011.

[29] N. Belghini, A. Zarghili, J. Kharroubi, and A. Majda, "Sparse random projection and dimensionality reduction applied on face recognition," in *The Proceedings of International Conference on Intelligent Systems & Data Processing*, 2011, pp. 78–82.

[30] J. E. Fowler, Q. Du, W. Zhu, and N. H. Younan, "Classification performance of random-projection-based dimensionality reduction of hyperspectral imagery," in *2009 IEEE International Geoscience and Remote Sensing Symposium*, vol. 5. IEEE, 2009, pp. V–76.

[31] K. Varmuza, C. Engrand, P. Filzmoser, M. Hilchenbach, J. Kissel, H. Krüger, J. Silén, and M. Trieloff, "Random projection for dimensionality reductionapplied to time-of-flight secondary ion mass spectrometry data," *Analytica chimica acta*, vol. 705, no. 1-2, pp. 48–55, 2011.

[32] A. Juvonen and T. Hamalainen, "An efficient network log anomaly detection system using random projection dimensionality reduction," in *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2014, pp. 1–5.

[33] A. Juvonen, T. Sipola, and T. Hämäläinen, "Online anomaly detection using dimensionality reduction techniques for http log analysis," *Computer Networks*, vol. 91, pp. 46–56, 2015.

[34] J. J. Amador, "Random projection and orthonormality for lossy image compression," *Image and Vision Computing*, vol. 25, no. 5, pp. 754–766, 2007.

[35] S. R. Oliveira and O. R. Zaiane, "Privacy-preserving clustering by object similarity-based representation and dimensionality reduction transformation," in *Proc. of the Workshop on Privacy and Security Aspects of Data Mining (PSADM04) in conjunction with the Fourth IEEE International Conference on Data Mining (ICDM04)*, 2004, pp. 21–30.

[36] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Transactions on knowledge and Data Engineering*, vol. 18, no. 1, pp. 92–106, 2005.

[37] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, 1960.

[38] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[39] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3329–3337.

[40] S. S. Gupta, "Probability integrals of multivariate normal and multivariate t1," *The Annals of mathematical statistics*, pp. 792–828, 1963.

[41] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[42] Wikipedia contributors, "Truncated normal distribution — Wikipedia, the free encyclopedia," 2019, [Online; accessed 12-June-2019]. [Online]. Available: https://en.wikipedia.org/wiki/Truncated_normal_distribution#Moments

[43] P. Li, M. Mitzenmacher, and M. Slawski, "Simple strategies for recovering inner products from coarse random projections," 2017.

[44] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 380–388.

[45] P. Li, M. Mitzenmacher, and A. Shrivastava, "Coding for random projections," in *International Conference on Machine Learning*, 2014, pp. 676–684.

[46] A. Ng and M. Jordan, "Y. weiss," *On spectral clustering: Analysis and an algorithm, NIPS*, vol. 14, pp. 849–856, 2002.

[47] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[48] K. Y. Yeung and W. L. Ruzzo, "Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[49] Y. Chen, K. W. Ng, and Q. Tang, "Weighted sums of subexponential random variables and their maxima," *Advances in applied probability*, vol. 37, no. 2, pp. 510–522, 2005.

[50] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20," Tech. Rep., 1996.

[51] L. Isserlis, "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables," *Biometrika*, vol. 12, no. 1/2, pp. 134–139, 1918.

[52] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[53] N. Fan and P. M. Pardalos, "Multi-way clustering and biclustering by the ratio cut and normalized cut in graphs," *Journal of combinatorial optimization*, vol. 23, no. 2, pp. 224–251, 2012.

[54] M. Slawski, "Stat 672 class 10: Clustering," https://mymasonportal.gmu.edu/bbcswebdav/pid-6702797-dt-content-rid-89070567_1/courses/18328.201710/class-10%281%29.pdf, April 2017.

[55] N. L. Johnson, S. Kotz, and N. Balakrishnan, "Continuous univariate distributions," 1994.

Curriculum Vitae

Glenn T. Hui graduated from the University of Toronto Schools, Toronto, Canada in 1999. He received his Bachelor of Science from the University of Toronto in 2004. He received his Master of Science from the London School of Economics and Political Science in 2005. He worked in government statistics before starting his Doctor of Philosophy at George Mason University in 2014.