Quantum Applications of an Efficient Solution to Compressive Phase Retrieval

Kunal Marwaha

University of California, Berkeley (UCB)

Abstract

Quantum state discrimination is at the heart of quantum information - when measurement inherently perturbs the system, decision and reconstruction problems about the state are crucial to building useful quantum-informational systems. When a state or process is easily reproducible, quantum tomography can be used, but it requires resources exponentially increasing with the system size [1]. Instead, compressive techniques can be applied to reconstruct a state vector $|\psi>$, especially in applications of quantum circuit and process verification, correction against randomized error, and protection from localized, adversarial noise. In each of these realms, there exist useful problems where one hopes to reconstruct a sparse state vector. Techniques from other fields, including information theory and signal processing, can be ported to quantum information: In particular, using a sparse-graph solution to the compressive phase retrieval problem [2], I develop a guaranteed quantum measurement construction process so that a K-sparse state vector can be asymptotically reconstructed in order-optimal decoding time $O(K)$. This quantum measurement construction process will work for any noise-robust, one-stage solution to the compressive phase retrieval problem. Practical tradeoffs between sampling size and average error are simulated and discussed.

Quantum state discrimination is the process of reconstructing or estimating a system $|\psi>$ or ρ. If some prior state information is known, the main approach for state discrimination uses quantum hypothesis testing, where a state is determined from a set of known possibilities [3]. This procedure is useful when a state is difficult or expensive to reproduce - even non-orthogonal states can be determined unambiguously [4]. However, when the state is easily reproducible, quantum tomography, and more recently, compressive quantum tomography, can reconstruct a state by clever, repeated measurement [1]. Compressive techniques are particularly useful: unlike standard quantum tomography, they do not necessarily require exponentially-growing physical resources with the number of sytem qubits, and instead take advantage of the assumed sparsity of the system [5]. Thus, developments in compressive approaches have inherent potential to improving current tomographical techniques.

Compressive sensing is an old field devoted to better understand limits of signal recovery, with widespread applications in physics and electrical engineering. The focus is on sparsity - can a complex signal known to be sparse in some domain be efficiently and/or fully reconstructed? If the signal is known or assumed to be sparse, optimal reconstruction algorithms should require fewer measurements. Developments within and nearby this field could have stark consequence within quantum information, where the dimension of a state vector grows exponentially with the number of system qubits.

A related, more useful problem has been studied under the term "phase retrieval" within signal processing, and will be the focus of this paper. Specifically, given a vector $x \in \mathbb{C}^N$, how many (and which) observation vectors $a_i \in \mathbb{C}^N$ are required to determine x from observations $y_i = |<a_i, x>|$, where $<a, b> = a^\dagger b$ denotes the Euclidean inner product on $\mathbb{C}^N$. This problem is challenging, due to magnitude-only measurements; non-linear approaches are necessary. This problem has significantly more approachable when x is assumed to be K-sparse, bearing the name "compressive phase retrieval". At least $4K - O(1)$ measurements are necessary [6] to reconstruct K values of x. Thus, for standard tomographical procedures (which don't exploit sparsity), at least $4N - O(1) = 4(2^n) - O(1)$ measurements are necessary - requiring exponentially more resources compared to system size.

One recent paper [2] has found an order-optimal solution to the compressive phase retrieval problem using sparse-graph codes, with $O(K)$ measurements and $O(K)$ decoding time required to completely reconstruct x with high probability. Concretely, one can use this algorithm and $14K$ measurements to determine a K-sparse vector x of any size, with

probability $1 - 10^{-7}$. Here, I use this algorithm in particular and unpack its possibilities.

Compressive phase retrieval has significant potential in three major quantum-informational applications: quantum circuit verification and process tomography, random-error correction, and protection against localized, adversarial noise. In quantum information, the "signal" is the state vector $|\psi>$, with dimension $N = 2^n$, where n is the number of qubits. In many of these use cases, it can be useful to imagine $|\psi>$ as sparse. In particular, state corruption (both random and adversarial) can be considered to be localized to just a few (unknown) qubits, and other quantum compressive techniques have been applied to nearly-sparse systems [5], well-approximated by sparse states. Thus, sparse, systematic errors can be determined by sending in a known separable state $|0_L>$ and efficiently reconstructing final (sparse) state $|\psi>$.

It is also known that quantum circuitry is universal - so, any sort of quantum gate could be imagined, including one that just adjusts a few qubits, or one that guarantees the state of most qubits. For example, consider a deterministic circuit C that operates on a 1000-qubit system, but alters at most 10 qubits. If this circuit is used on a trial state with all qubits initialized to 0 in a computational basis, the final state would have at most 10 qubits on. The state vector $|\psi>$ has dimension $N = 2^{1000} \approx 10^{300}$, but the all possible state collapses of $|\psi>$ have at most 10 qubits collapsed as 1. Thus, the sparsity of $|\psi>$ is at most $K = \sum_{i=0}^{10} \binom{1000}{i} < 10^{24}$. Characterization and verification of this circuit is orders-of-magnitude more efficient with techniques exploiting sparsity.

Compressive phase retrieval has the possibility to revolutionize a very similar technique: Quantum Process Tomography. With a similar problem setup to quantum circuitry, an unknown time-evolution can be characterized using tomography of a reproducible state. In the same way, compressive techniques outperform standard quantum tomography as the system size increases [5].

Of course, challenges exist when porting anything to a quantum-informational setting. In particular, the No-Cloning Theorem dictates that states can only be transferred, not copied. This limits the scope of quantum tomography and quantum compressive phase retrieval: If a decoding process needed multiple measurements of a single state, one could not copy numerous times before decoding. Instead, access to a reproducible state (or consistent process resulting in a reproducible state) is mandatory. In usual physics, this can be a simple light source - in quantum information, it must be a pipeline converting an initial

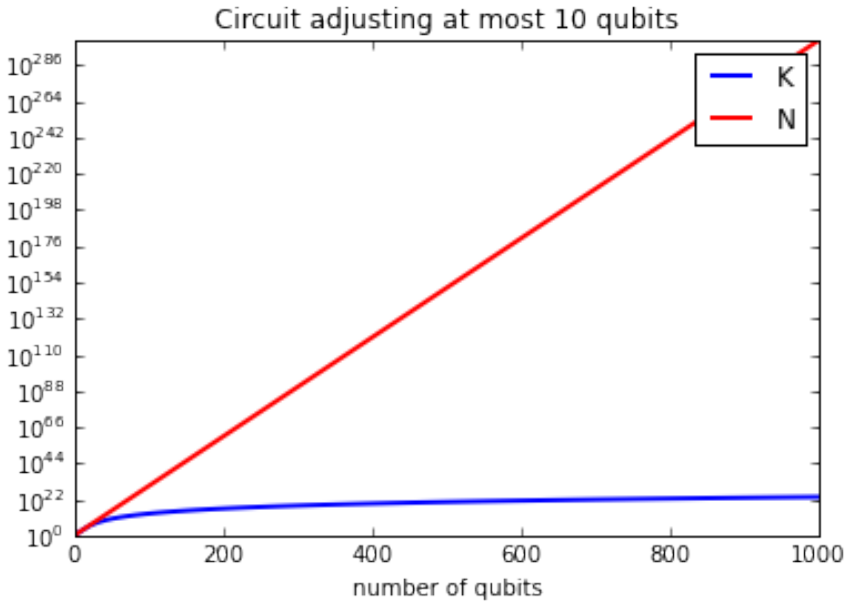

FIG. 1. Using example circuit C and initial state $|0_L>$. When measured along the computational basis, all collapse outcomes of the final state $|\psi>$ have at most 10 qubits on, limiting the sparsity K to be at most $\sum_{i=0}^{10} \binom{n}{i}$. The sparsity of $|\psi>$ grows polynomially with the number of qubits n (asymptotically, $K \propto n^{10}$) but the dimension $N = 2^n$ of the state vector grows exponentially.

$|0_L>$ to $|\psi>$.

Additionally, quantum collapse limits the amount of useful information from each state. On first pass, a state holding $N = 2^n$ values only returns n bits after direct measurement along the computational basis. More clever approaches can re-use states (repeated weak measurement), use Bayesian estimation strategies [3][7], or use a series of measurement operators P_i not on the computational basis (enforcing that $\sum_i P_i^\dagger P_i = I$). All approaches to integrate compressive phase retrieval must overcome these challenges.

Below is a brief discsussion of the best known compressive phase retrieval solution, by Pedarsani, Lee, Ramchandran. The reader is cautioned that due to magnitude-only measurements, the equations are non-linear: usual tricks may not apply. A more complete description and analysis is in their paper [2]. They model the classical problem with a bipartite graph matching values of $x_j, 1 \leq j \leq n$ to observations $y_i, 1 \leq i \leq M$. Each left edge of an observation y_i determines the observation vector a_i, measuring a certain subset of vector elements. Reconstructing the state vector x involves solving a system of equations

$y = |Ax|$, where each row of matrix A is the observation vector a_i.

The reconstructing code (decoder) relies on a method of singleton-coloring - if an observation is connected to a single nonzero vector element (thus dubbed "singleton"), then its nonzero vector element can be eliminated (or "colored") from the system of equations. If other vector elements are connected to this observation, they are exactly zero, further determining x and simplifying the equations. If singletons are common and "easy" to find, the system can be solved more efficiently than with traditional methods [2].

Of course, this only works if the observations were complex numbers, i.e. if there was no loss in system measurement. To compensate, each observation a_i is physically conducted by four (4) trigonometric observation vectors, triangulating up to a global phase. Thus, $m = 4M$ measurements are truly required to reconstruct x.

The main contribution by Pedarsani, Lee, Ramchandran was to invent a random construction process of a_i's where singletons can be efficiently determined and "colored", interleave a clever triangulation to compensate for magnitude-only observations, and conduct a preliminary analysis of the success of their family of algorithms, dubbed "PhaseCode". Their first algorithm, using regular graphs, has high levels of success with $O(K)$ measurements, and solves the system of equations in $O(K)$ time. Moreover, this algorithm has been shown to be experimentally successful in at least one setting [8].

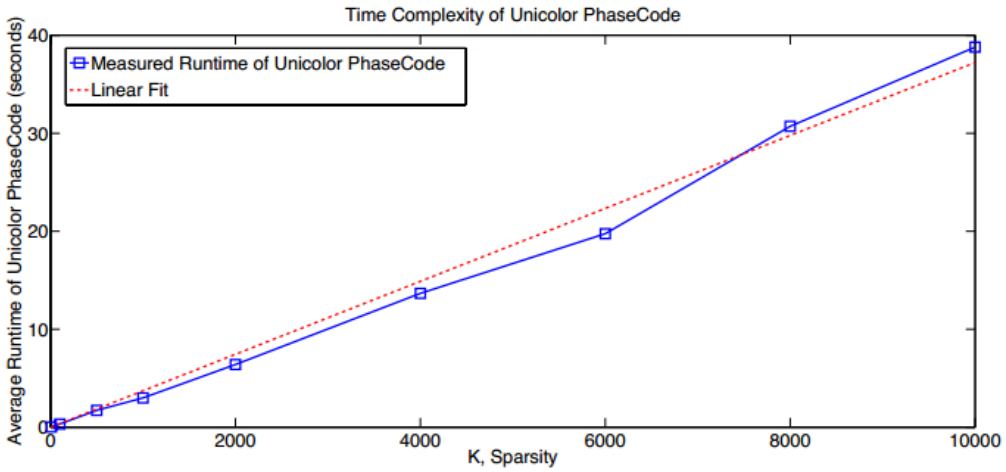

FIG. 2. From Pedarsani, Lee, Ramchandran [2]. The measured runtime of their compressive phase retrieval algorithm is linear with K (sparsity), and seemingly independent of N (dimension).

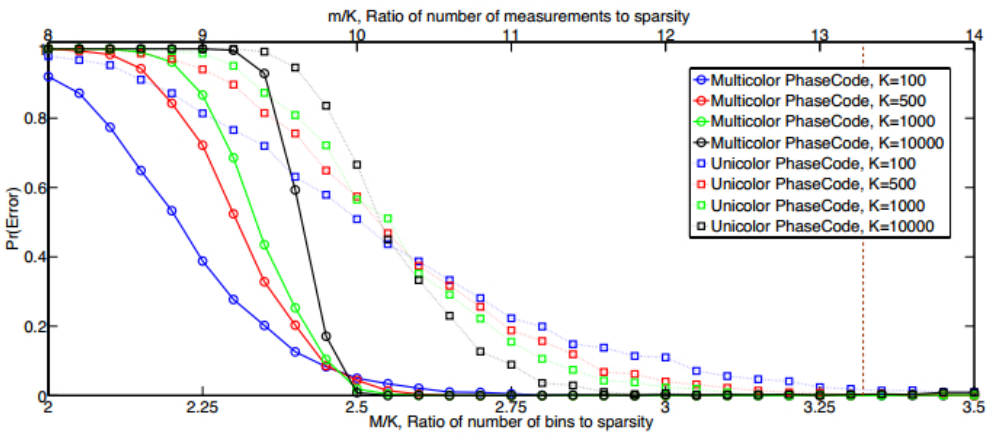

FIG. 3. From Pedarsani, Lee, Ramchandran [2]. Using $m = 4M$ measurements, probability of successful reconstruction nears unity as M nears $3.5K$ ($m = 14K$ measurements). In particular, using $M = 3.32K$ and the third column of the table, they measure 3.2×10^{-6} chance of error.

The task, now, is to use this algorithm in a quantum mechanical setting. The simplest conversion is to convert each of the m observation vectors a_i to some measurement operator P_i. I do so by creating one quantum measurement $P = \{P_i\}$, where the first m measurement operators are chosen to be $P_i = \alpha_i a_i a_i^\intercal$. In order to enforce $\sum_i P_i^\dagger P_i = \sum_i F_i = I$, I add one measurement operator, P_{m+1}, such that $P_{m+1}^\dagger P_{m+1} = F_{m+1} = I - \sum_{i=1}^m F_i$. The quantum measurement P then has $m+1$ possibilities, with each index i occuring with probability $\|P_i x\|^2$. Thus, the observation $|<a_i, x>| = \frac{\|P_i x\|}{\alpha_i \|a_i\|}$ can be estimated by repeated sampling via measurement P.

Such a scheme is always possible. Normalization constants $\alpha_i \in \mathbb{R}$ are chosen to enforce positive-semidefiniteness of F_{m+1}. A constructive proof is below (done by the author):

If the eigenvalues $\sigma(A)$ of matrix A are $\{\lambda_i\}$, then the eigenvalues of $I - A$ are $\{1 - \lambda_i\}$. (This is straightforward, since A and $I - A$ share eigenvectors: If $(A - \lambda I)v = 0$, then $(-A - (-\lambda)I)v = 0$, so $((I - A) - (1 - \lambda)I)v = 0$.) Additionally, for Hermitian matrices H, the maximum eigenvalue λ_{max} is the maximum value of $<v, Hv> = v^\dagger H v$ given a normalized vector v; in other words, $\lambda_{max}(H) = sup_{\|v\|=1} v^\dagger H v$. Thus, a triangle inequality holds: $\lambda_{max}(H_1 + H_2) \leq \lambda_{max}(H_1) + \lambda_{max}(H_2)$, with equality only when eigenvectors are shared. Since $F_i = P_i^\dagger P_i$, matrices $F_i (1 \leq i \leq m)$ are guaranteed to be positive semidefinite (and its eigenvalues guaranteed to be real). Here, $P_i = \alpha_i a_i a_i^\intercal$, so $F_i = a_i^\star a_i^\dagger \alpha_i^\star \alpha_i a_i a_i^\intercal = \alpha_i^2 \|a_i\|^2 a_i a_i^\dagger$.

By the above, $\lambda_{min}(F_{m+1}) = 1 - \lambda_{max}(\sum_{i=1}^{m} F_i)$, which must be greater than $1 - \sum_{i=1}^{m} \lambda_{max}(F_i)$ by the triangle inequality. Choosing $\alpha^2 < m^{-1}\|a_i\|^{-2}\lambda_{max}(a_i a_i^\dagger)^{-1}$ enforces that $\lambda_{min}(F_{m+1}) \geq 1 - \sum_{i=1}^{m} \lambda_{max}(F_i) > 1 - \sum_{i=1}^{m} \frac{1}{m} = 0$, guaranteeing all eigenvalues of F_{m+1} to be positive. Thus, it is possible to construct a positive definite F_{m+1}, which, via Cholesky decomposition, guarantees an operator P_{m+1} (and thereby a proper quantum measurement $P = \{P_j\}, 1 \leq j \leq m+1$) from any set of observation vectors $a_i, 1 \leq i \leq m$.

Simulations are conducted to estimate observations $O_i = |<a_i, x>|$. In this setting, the m measurements conducted by observation vectors a_i are converted into one (1) quantum measurement P. Here, repeated sampling is required to estimate the probability $\|P_i x\|^2$ of each index i occurring. Theoretically, infinite samples are required to noiselessly estimate each observation $O_i = |<a_i, x>| = \frac{\|P_i x\|}{\alpha_i \|a_i\|}$. By the CLT, l samples of an binary indicator with probability p will approximate as $\hat{p} \sim \mathcal{N}(p, \frac{p-p^2}{l})$. Thus, repeated sampling using P will approximate each observation as $\hat{O_i^2} \sim \mathcal{N}(O_i^2, \frac{\|P_i x\|^2 - \|P_i x\|^4}{\alpha_i^4 \|a_i\|^4 l})$.

A standard simulation is conducted as follows: Using m observation vectors from the PhaseCode algorithm, construct the $m+1$ quantum measurement operators in the scheme described above. Then, measure l samples: For each sample, randomly select one of the measurement operators in accordance with its probability $\|P_i x\|^2$. For each index i, estimate its occurrence probability $\widehat{\|P_i x\|^2}$ as the number of selections of outcome i per sample size l. Thus, observation i is estimated as $\hat{O_i} = \frac{\sqrt{\widehat{\|P_i x\|^2}}}{\alpha_i \|a_i\|}$. Relative error for outcome i is computed as $\frac{|O_i - \hat{O_i}|}{O_i}$, and average relative error of nonzero observations (ARE) is computed.

Because the simulated scheme guarantees a proper quantum measurement (by guaranteeing positive-definiteness of $F_{m+1} = P_{m+1}^\dagger P_{m+1}$), it may not be the most effective for finite sampling. In particular, simulations consistently had outcome $m+1$ occurring more than 99% of the time - vastly increasing the number of required samples for precise measurements. (Since many outcomes had occurrence probability less than .001, more samples are required to achieve granularity in frequency counts across outcomes.) This leads to higher-than-optimal ARE per sample.

This process reduces the importance of the number of observation vectors a_i needed to reconstruct x - this simply adjusts the number of possibilities of quantum measurement P. Instead, the important metric is decoding time - as the system size (i.e. n, number of qubits) grows, the number of equations needed to reconstruct an arbitrary x grows exponentially with n, but algorithms that take advantage of sparsity in x can do much better. In particular,

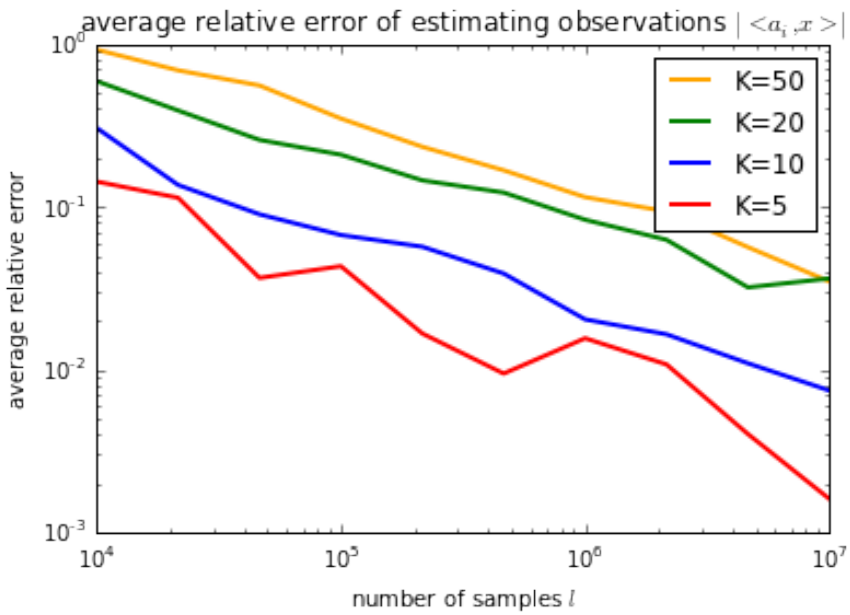

FIG. 4. Simulated average relative error of quantum measurement P as compared to classical $y = |Ax|$ of a K-sparse state vector. Approximately, $ARE \propto l^{-2}$, where ARE is the average relative error over nonzero observations. In this image, the dimension $N = 128 = 2^7$.

PhaseCode algorithms have decoding time $O(K)$ - independent of system size! Thus, for arbitrarily large systems, sparse vectors can be reconstructed efficiently.

There is some caveat - noiseless PhaseCode algorithms have decoding time $O(K)$, but it is impossible to theoretically achieve noiseless observation without infinite sampling. Practically, however, there may be a happy tradeoff between l and runtime - in fact, there may be an lower bound on l such that PhaseCode algorithms will still work with high probability. Preliminary simulations suggest such a possibility for large K; future construction-process adjustments and analysis may help discern this tradeoff.

Additionally, noisy PhaseCode algorithms exist, with preliminary algorithms using $O(K \log N)$ measurements and decoding with efficient (non-exponential) runtime [2]. There may be future robust PhaseCode algorithms able to handle this noise. In general, compressive phase retrieval solutions, robust to some noise, can be applied to quantum information in the above construction process from a_i to P. Even multi-stage solutions, where future a_i depend on previous a_i, can be constructed using multiple quantum measurements P_j. For quantum information, there may be more practical or realizable compressive phase retrieval

solutions than PhaseCode - even if so, this process is potentially useful. Future work should be devoted to practical tradeoffs and process improvements - what is the best quantum compressive technique that minimizes both sample size l and decoding time?

Thus, developments in compressive phase retrieval could be transformative for quantum information; as systems scale, a replacement reconstruction tool for quantum tomography is required. Even with limitations such as the No-Cloning Theorem, compressive techniques have possibilities in verifying quantum circuits and processes, protecting from interference, and estimating systematic and localized error. Solutions from other research fields may be ripe for quantum information. Here, I have used an information-theoretic solution to the compressive phase retrieval problem to develop an asymptotically efficient quantum compressive reconstruction process. This decoding process is as fast as the phase-retrieval solution: order-optimal time $O(K)$. Preliminary simulations show that although some estimation error exists, it decreases with sample size l - there may be some happy practical tradeoff with noiseless algorithms given high l. Analysis of noise-robust algorithms may be required in the future for this result be practically useful. The quantum measurement construction process may also have other practical improvements to reduce estimation error under finite l. Overall, quantum compressive state reconstruction techniques provide a friendly space to pair computer science theory research with relevant applications.

The author would like to acknowledge Ramtin Pedarsani, Kangwook Lee, and Kannan Ramchandran for willingness to discuss their algorithm and its quantum implications; Kevin Young, Mohan Sarovar, and Birgitta Whaley for a formal course in quantum information; Rostam Razban for motivation and healthy discussion; and Daniel Freeman for guidance connecting the two worlds of information theory and quantum physics.

[1] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Phys. Rev. Lett. **105** (2010), 10.1103/physrevlett.105.150401

[2] R. Pedarsani, K. Lee, and K. Ramchandran, CoRR **abs/1408.0034** (2014)

[3] A. Chefles, Contemporary Physics **41**, 401 (2000)

[4] R. Clarke, A. Chefles, S. Barnett, and E. Riis, Physical Review A **63** (2001), 10.1103/physreva.63.040305

[5] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White, Phys. Rev. Lett. **106** (2011), 10.1103/physrevlett.106.100401

[6] T. Heinosaari, L. Mazzarella, and M. M. Wolf, Communications in Mathematical Physics **318**, 355 (2013)

[7] G. Kimura, T. Miyadera, and H. Imai, Physical Review A **79** (2009), 10.1103/physreva.79.062306

[8] M. Mirhosseini, O. S. Magaña-Loaiza, S. M. H. Rafsanjani, and R. W. Boyd, Phys. Rev. Lett. **113** (2014), 10.1103/physrevlett.113.090402

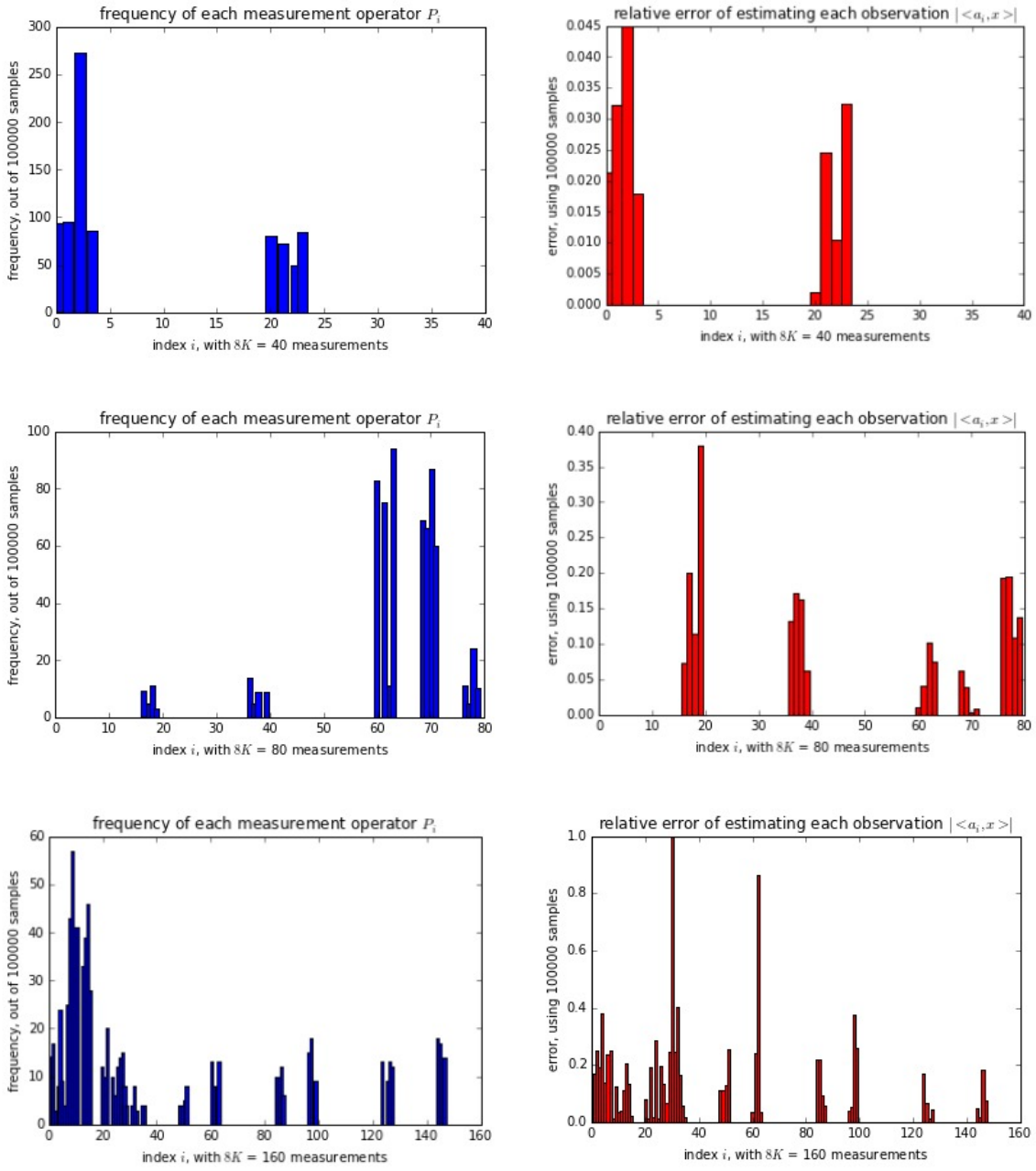

FIG. 5. Simulation plots of frequency of each measurement outcome and relative error of each observation $\frac{\hat{O}_i - O_i}{O_i}$ with $K = 5, 10, 20$. In this schema, F_{m+1} was chosen to guarantee positive-definiteness, which causes a high occurrence probability (typically $> 99\%$), reducing granularity of other outcomes and increases the error. (Outcome $m+1$ is not pictured in the graphs, as it is not used to reconstruct x). More clever choices of F_{m+1} could ameliorate this issue. If the relative error of observation i is 1.0, then the quantum measurement $\hat{O}_i$ incorrectly output 0 (also caused by low granularity of outcomes $1 \leq i \leq m$. In this image, $l = 10^5$ samples and the dimension $N = 128 = 2^7$.

```
##########################Quantum PhaseCode Simulations#########################
#using Kangwook Lee's code-matrix generating code:
##build measurement matrix A
##construct quantum measurement operators P_i
##compare differences between |Ax| and quantum measurement

import matplotlib.pyplot as plt
import random
from __future__ import division
j = complex(0,1)

####set parameters here:
n = 150 #n is dimension
k = 5 #k is sparsity
m = 2*k #m is number of code-matrix measurements (in total, 4m measurements)
l = 100000 #l is number of samples used in repeated quantum measurement

#####create k-sparse state vector, with random complex values
onbits = random.sample(range(n), k)
x3 = []
for i in range(n):
    if i in onbits:
        x3.append(complex(rand(1)[0], rand(1)[0]))
    else:
        x3.append(0)
x3 = (x3 / np.linalg.norm(x3)) #normalize vector
# print x3

####create trigonometric matrix (4xn, to triangulate, as discussed in PhaseCode paper)
omega = 0.5*pi / n
omega_prime = 0.25*pi / n #2*pi*rand(n-1)[0]/n is the specification, but Kangwook Lee's simulations set w'=0.5w
trig_matrix = []
for i in range(n):
    power = j*omega*(i+1)
    trig_matrix.append([exp(power), exp(0-power), 2*cos(omega*(i+1)), exp(j*omega_prime*(i+1))])
# print trig_matrix

####create code matrix (this function is discussed in PhaseCode paper)
code_matrix = matrix_generator().generate_G_irregular(m, n)
# print code_matrix

####row-tensor product matrices, as per PhaseCode paper
def row_tensor_product(trig_matrix, code_matrix, m):
    A = []
    for i in range(len(trig_matrix)):
        col = []
        for k in range(m):
            if k in code_matrix[i]:
                col += trig_matrix[i]
            else:
                col += [0,0,0,0]
        A.append(col)
    return A
A = row_tensor_product(trig_matrix, code_matrix, m)

####switch from row-major to column-major
def row_to_col(A):
    B = []
    for i in range(len(A[0])):
        B.append([])
        for j in range(len(A)):
            B[i].append(A[j][i])
    return B
A = row_to_col(A)
```

FIG. 6. Simulation code attached for reference.

```
#####Measurement!
#####classical approach: y = |Ax|
y = np.absolute(np.dot(A, x3)) # decoder can reconstruct x from code_matrix and y

####quantum approach: create measurement operators 1<=k<=4m
M = [] #list of operators P_i
M2 = [] #list of operators F_i
alphas = [] #squares of normalization constants (\alpha^2)
norms = [] #norms of observation vectors
for item in A:
    norm = np.linalg.norm(item)
    new_mat = np.outer(item, item)
    new_mat_sq = np.outer(np.dot(norm, map(conjugate, item)), np.dot(norm, item))
    l_max = max(np.linalg.eigvalsh(new_mat_sq))
    alphasq = (1/(4.01*m*l_max)) #4.01 for fudge factor (instead of 4) to ensure positive definiteness
    M.append(np.dot(pow(alphasq, 0.5), new_mat))
    M2.append(np.dot(alphasq, new_mat_sq))
    alphas.append(alphasq)
    norms.append(norm)

####quantum approach: create measurement operator 4m+1
#by triangle inequality, last_matrix will be positive definite
last_matrix = np.eye(n)
for item in M2:
    last_matrix = last_matrix - item
M2.append(last_matrix)
M.append(np.linalg.cholesky(last_matrix).T.conj())
alphas.append(1)
norms.append(1)

####sanity check: classically multiply measurement operators to obtain |P_i x|^2
probs = [] #list of expected probabilities of outcomes
output = [] #should match classical y=|Ax|
for i in range(len(M)):
    num =  np.linalg.norm(np.dot(M[i], x3))
    corrected = num*sqrt(1/alphas[i])/norms[i]
    probs.append(pow(num, 2))
    output.append(corrected)
    #all error is consistently <10^-15 (due to rounding)

####histogram: estimate probability |P_i x|^2 to estimate observations
h = np.random.choice(range(4*m + 1), size=l, p=probs)
h = filter(lambda x: x != 4*m, h)
out2 = [] #should be near classical y=|Ax|
error = [] #track magnitude of relative error between classical and quantum measurements
for i in range(4*m):
    pr2 = len(filter(lambda x: x == i, h)) / l
    out2.append(sqrt(pr2)*sqrt(1/alphas[i])/norms[i])
    if y[i]:
        error.append(abs((y[i]- out2[i]) / y[i]))
#         print out2[i], output[i], y[i]; print "percent error:", abs((y[i]- out2[i]) / y[i])
    else:
        error.append(0)
```

FIG. 7. Simulation code attached for reference.