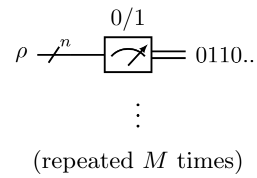
Tech Note T21: Estimating observables from sampled bit strings

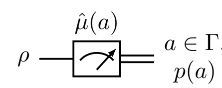
Note by Zlatko Minev, for John Watrous (2022-03)

Background. An experiment on a quantum computer results in classical data. The experiment samples nominally-identical copies of the same state ρ of the system some number of times $M \in \mathbb{Z}_{\geq 0}$. For an n-qubit system, each of the M samples returns a single (classical) binary string, such as $0110\ldots$, sampled from a distribution determined by ρ and the measurement basis. We call the empirical distribution of the M sampled bitstrings *the counts*, which is a stochastic integer vector over the set of all 2^n possible measurement outcomes, i.e., the binary strings of length n, denoted $\Sigma = \{00\ldots00, 00\ldots01, \ldots, 11\ldots11\}$. The classical counts data is post-processed to compute some desired statistic, such as the expectation of some operator $\left\langle O \right\rangle$; e.g., some Pauli string $\langle ZIZI\ldots\rangle$.

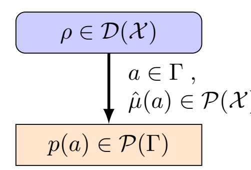


Question. Given the sampling error, how close to true is our estimate of the derived statistic from the bitstrings? How does the sampling error in the bitstrings propagate through the computation? How many M shots should an experimentalist run to obtain a certain confidence?

Measurement setup. Consider an n-qubit register over the binary alphabet $\Sigma = \{0,1\}^{\otimes n} = \mathbb{Z}_2^n$, with the associated complex Euclidian space $\mathcal{X}$. The register is characterized by a state $\rho \in \mathcal{D}(\mathcal{X})$ **?**, Sec. 2.1.2. In general, for a measurement with possible outcomes a drawn from some alphabet Γ, each measurement outcome is associated with a measurement operator $\mu(a) \in \mathrm{Pos}(\mathcal{X})$; i.e., using the mapping is $\mu\colon \Gamma \in \mathrm{Pos}(\mathcal{X})$ **?**, Eq. (2.199). The probability vector over the possible measurement outcomes is[1]



$$p(a) = \langle \mu(a), \rho \rangle \in \mathbb{R} \;, \qquad \text{with } 0 \leq p(a) \leq 1 \text{ and } \sum_{a\in\Sigma} p(a) = 1 \;. \tag{1}$$



Measuring in the computational basis, $\Gamma = \Sigma$, the case of interest here, the measurement operators are simply the projectors on the computational basis $\mu(a) = |a\rangle\langle a|$. The probability to sample the oucome a is $p(a) = \mathrm{Tr}(|a\rangle\langle a|\rho) = \langle a|\rho|a\rangle =: \rho_{aa}$, the a-th diagonal entry of the state. The size of the outcome space grows exponentially $|\Sigma| = 2^n$.

General observable. Without loss of generality,[2] an observable *diagonal* in the computational basis can be written as

$$O = \sum_{a\in\Sigma} O(a)\,|a\rangle\langle a| = \sum_{a\in\Sigma} O(a)\,\mu(a) \;, \tag{2}$$

with $O : \Sigma \in \mathbb{R}$. The measurement operator associated with O and outcome outcome a and is $O(a)\,\mu(a)$, merely scaled version of the computational basis operator $\mu(a)$. Assuming orthogonal μs, $O(a) = \frac{\langle \hat{\mu}(a), \hat{O} \rangle}{\langle \hat{\mu}(a), \hat{\mu}(a) \rangle}$.

Pauli-Z observables and the Walsh-Hadamard transform. For Pauli-Z observables, such as $IZZI\ldots$, $O(a) \in \{-1, +1\}$. The set of all Pauli Z observables is $\mathcal{P}_Z = \left\{Z^b : b \in \Sigma\right\}$, where Z^b is understood to be the n-qubit Pauli operator with I and Z Paulis in the slots where the n-qubit binary string b has 0 and 1 elements, respectively; i.e., $Z^{00110} = IIZZI$. There are 2^n bitstrings and 2^n Pauli-Z operators; i.e., $|\mathcal{P}_Z| = |\Sigma| = \dim \mathcal{X} = 2^n$.

The two sets are informationally equivalent in that there is a bijective mapping from one operator set to the other, given by the Walsh-Hadamard transform:[3]

Bit string y

	$[\langle x, y\rangle]_2$	00	01	10	11
	00	0	0	0	0
X	01	0	1	0	1
	10	0	0	1	1
	11	0	1	1	0

[1] The variable $p(a) = \langle \hat{\mu}(a), \rho \rangle$ is strictly real because the inner product of two Hermitian operators is real.
[2] Other observables can be rotated to this basis.
[3] See calculation (C58) in my hand-written notes.

$$Z^b = \sum_{a\in\Sigma} (-1)^{\langle b,a\rangle} |a\rangle \langle a| \ , \qquad \langle b,a\rangle \coloneqq \sum_{i=0}^{n-1} b_i a_i \ , \tag{3}$$

$$|a\rangle \langle a| = \frac{1}{|\Sigma|} \sum_{b\in\Sigma} (-1)^{\langle a,b\rangle} Z^b \ , \qquad \Big\langle |a\rangle \langle a| \,, Z^b \Big\rangle = (-1)^{\langle a,b\rangle} \ , \tag{4}$$

where the simple inner product of the binary words $\langle b,a\rangle$ and its sum are ideally to understood modulo 2; i.e., the sum is $\oplus$ mod 2 and $\langle b,a\rangle \in \mathbb{Z}_2$.[4] Recasting Eq. (3) in the language of Eq. (2), $O(a) = (-1)^{\langle a,b\rangle}$. The kernel element of the transformation can be understood as the (a,b)-th entry of the Walsh-Hadamard matrix $W_{ab} = (-1)^{\langle a,b\rangle}$; note, the inverse is $\left(W^{-1}\right)_{ab} = \frac{1}{|\Sigma|} W_{ab}$.

Define the vectors of ideal expectations $q_Z^{\mathrm{ideal}}(b) = \left\langle Z^b \right\rangle$ and $q_0^{\mathrm{ideal}}(a) = \langle |a\rangle \langle a| \rangle$ over all the Pauli and computational-basis projection operators, respectively. The two are related by the Walsh- Hadamard transform

$$q_Z^{\mathrm{ideal}} = W q_0^{\mathrm{ideal}} \ . \tag{5}$$

21.1 Sampling

A single measurement in the computational basis of an n-qubit register yields an outcome drawn from the alphabet $\Sigma = \mathbb{Z}_2^n$ according to the probability vector $\{p(a) : a \in \Sigma\}$, which is a general probability and related to ρ by Eq. (1). The only constraint on $p(a)$ is $\|p(a)\|_1 = 1$.

In an experiment, we perform M independent trials, each of which leads to 1 of the 2^n possible outcomes. The total counts realized is a random vector $\{C(a) : a \in \Sigma\}$, where $C(a)$ obeys a multinomial distribution. The sum of the counts is $\|C(a)\|_1 = M$, an element's possible values are $C(a) \in \{0, \ldots, M\}$. Using the notation, $p(a) \in \ p_a$, $C(a) \in \ C_a$, etc., the expectation and variance are $\mathbb{E}[C_a] = Mp_a$ and $\mathbb{V}[C_a, C_b] = Mp_a(\delta_{ab} - p_b)$, respectively.

To convert the counts to ... WIP.

[4]Since this is the exponent of -1, it doesn't really matter here.