Sampling Probability Distributions à la Grover

Robert R. Tucci
P.O. Box 226
Bedford, MA 01730
tucci@ar-tiste.com

October 11, 2010

In this blog post, I will describe Grover's sampling algorithm, an algorithm given in Ref.[1], for sampling probability distributions using a quantum computer. I will also compare very briefly Grover's sampling algorithm with the sampling algorithm used by my computer program Quibbs.

I will use my customary notation. Recall that I like to use $Bool = \{0, 1\}$, $P_0 = |0\rangle\langle 0|$, $P_1 = |1\rangle\langle 1|$. Also $N_S = 2^{N_B}$ is the number of states for N_B bits. Qubit positions will be labeled by Greek letters.

Consider N_B qubits labeled $\vec{\alpha} = (\alpha_{N_B-1}, \ldots, \alpha_1, \alpha_0)$ and a single auxilliary qubit labeled β. Suppose $|\Psi\rangle$ is a state of the $N_B + 1$ qubits $(\vec{\alpha}, \beta)$ and let $P_0(\beta) = |0\rangle_\beta\langle 0|_\beta$. Define an operator G by

$$G = -(-1)^{|\Psi\rangle\langle\Psi|}(-1)^{P_0(\beta)} . \tag{1}$$

Grover's sampling algorithm involves calculating $G^M|\Psi\rangle$ for some suitable integer M. Note that if we define a unit vector $|\Psi_0\rangle$ by

$$|\Psi_0\rangle = \frac{P_0(\beta)|\Psi\rangle}{\sqrt{\langle\Psi|P_0(\beta)|\Psi\rangle}} , \tag{2}$$

then

$$P_0(\beta)|\Psi\rangle = (|\Psi_0\rangle\langle\Psi_0|)|\Psi\rangle , \tag{3}$$

and

$$P_0(\beta)|\Psi_0\rangle = (|\Psi_0\rangle\langle\Psi_0|)|\Psi_0\rangle . \tag{4}$$

Hence, $P_0(\beta)$ equals $|\Psi_0\rangle\langle\Psi_0|$ when acting on space $span(|\Psi\rangle, |\Psi_0\rangle)$. If we define

$$\tilde{G} = -(-1)^{|\Psi\rangle\langle\Psi|}(-1)^{|\Psi_0\rangle\langle\Psi_0|} , \tag{5}$$

then

$$G^M|\Psi\rangle = \tilde{G}^M|\Psi\rangle \ . \tag{6}$$

We see that Grover's sampling algorithm uses the original Grover's "search" algorithm with a starting state $|s'\rangle = |\Psi\rangle$ and a target state $|t\rangle = |\Psi_0\rangle$.

Grover's sampling algorithm uses for $|\Psi\rangle$ the state

$$|\Psi\rangle = \frac{1}{\sqrt{N_S}} \sum_{\vec{x}\in Bool^{N_B}} \left(\sqrt{Pr(\vec{x})}|0\rangle_\beta + \sqrt{1-Pr(\vec{x})}|1\rangle_\beta\right)|\vec{x}\rangle_{\vec{\alpha}} \ , \tag{7}$$

where $Pr()$ is the probability distribution that we wish to sample. For this $|\Psi\rangle$, one gets

$$|\Psi_0\rangle = \sum_{\vec{x}\in Bool^{N_B}} \sqrt{Pr(\vec{x})}|0\rangle_\beta|\vec{x}\rangle_{\vec{\alpha}} \ . \tag{8}$$

To build the state $|\Psi\rangle$ of Eq.7, one can proceed as follows. Define operators U and Γ by

$$U = \Gamma(\beta,\vec{\alpha})H^{\otimes N_B}(\vec{\alpha}) \ , \tag{9}$$

$$\Gamma(\beta,\vec{\alpha}) = \exp\left\{-i\sigma_Y(\beta)\sum_{\vec{x}\in Bool^{N_B}} \arccos(\sqrt{Pr(\vec{x})})P_{\vec{x}}(\vec{\alpha})\right\} \tag{10a}$$

$$= \sum_{\vec{x}\in Bool^{N_B}} \exp\left\{-i\sigma_Y(\beta)\arccos(\sqrt{Pr(\vec{x})})\right\} P_{\vec{x}}(\vec{\alpha}) \tag{10b}$$

It follows that

$$U|0^{N_B}\rangle_{\vec{\alpha}}|0\rangle_\beta = \tag{11a}$$

$$= \frac{1}{\sqrt{N_S}} \sum_{\vec{x}\in Bool^{N_B}} \exp\left\{-i\sigma_Y(\beta)\arccos(\sqrt{Pr(\vec{x})})\right\}|\vec{x}\rangle_{\vec{\alpha}}|0\rangle_\beta \tag{11b}$$

$$= |\Psi\rangle \ . \tag{11c}$$

Thus,

$$|\Psi\rangle = U|0^{N_B+1}\rangle \ , \tag{12}$$

and

$$(-1)^{|\Psi\rangle\langle\Psi|} = U \ \ (-1)^{|0^{N_B+1}\rangle\langle 0^{N_B+1}|} \ \ U^\dagger \tag{13}$$

The operator Γ is what I like to call a $U(2)$ multiplexor with N_B controls. It can be expanded into multiply controlled NOTS and single-qubit rotations using the computer program "Multiplexor Expander", described in Ref.[2].

Some drawbacks of Grover's sampling algorithm are as follows:

- Grover's sampling algorithm uses the full probability distribution $Pr(x)$ whereas Quibbs uses only conditional probabilities of $Pr(x)$. All the classical MCMC (Markov Chain Monte Carlo) methods also use only conditional probabilities.
- In general, for arbitrary $Pr()$, the operator Γ cannot be compiled, either exactly or to a good approximation, with polynomial efficiency (i.e., it cannot be decomposed into a SEO (Sequence of Elementary Operations) whose length scales polynomially in N_B).

References

[1] Lov Grover, "Rapid Sampling Through Quantum Computing" ariv:quant-ph/9912001,

[2] R.R. Tucci, "Code Generator for Quantum Simulated Annealing" by R.R. Tucci, arXiv:0908.1633