Beyond BBBV theorem, or not? A story of function oracles

Minseong Kim

July 28, 2023

Some background story

Six months ago, there was a person sipping coffee, annoyed by the following.

Suppose you have binary oracle (computational basis encoding) O_b that encodes real-valued function $f(t)$ for each index t. That is, $O_b|t\rangle|0\rangle = |t\rangle|f(t)\rangle$.

We know that for analog oracle O_a, which encodes function $f(t)$ as a superposition of function values, $O_a|0\rangle = \sum_t f(t)|t\rangle$, quantum Fourier transform (QFT) can be utilized - which is, sans NISQ computing issues, quite efficient. But this is not available for the binary oracle.

Some background story (2)

Now why is this so problematic? Well, because some function manipulations are best done in the analog oracle. For example, we may think of swapping frequencies (Fourier basis) of function (one case is such that $e^{it} + 2e^{i2t}$ becomes $2e^{it} + e^{i2t}$). But this frequency swap is a heavily quantum operation, which cannot be done with the binary oracle because it is a semiclassical operation. (The word locality is sometimes used.)

arXiv:1711.00465 provides some analysis into three types of oracle for encoding a function - binary, phase and probability. Now binary oracle was already introduced, so I will discuss slightly more into phase and probability oracles.

Binary and phase oracles

So given binary oracle O_b, we can utilize the simple phase kickback procedure such that we obtain phase oracle O_{ph}, which implements function as $O_{ph}|t\rangle = e^{if(t)}|t\rangle$. Basically, gate costs in converting O_b into O_{ph} are close to being nothing. **So conversion $O_b \rightarrow O_{ph}$ is easy.**

The problem is converting from O_{ph} to O_b. **This is not easy**, as we are condemned to some form of phase estimation! And phase estimation requires query complexity of $O(2^b)$, where b is the number of precision bits.

Binary, probability and analog oracles

Binary oracle O_b can be easily converted to probability oracle O_{pb}, which implements function $f(t)$ by $O_{pb}|t\rangle|0\rangle = |t\rangle(f(t)|0\rangle + \sqrt{1-f(t)}|1\rangle)$. This is easy: use computed value of $f(t)$ for the rotation degree between $|0\rangle$ and $|1\rangle$ and un-compute $f(t)$.

Now we obtain analog oracle O_a when we postselect on $|0\rangle$. If we do not assume away postselection, then we require something like amplitude amplification to convert from phase to analog oracle. **This can be very expensive depending on function** $f(t)$.

Phase and probability oracles

As seen in arXiv:1711.00465, depending on function $f(t)$, converting O_{ph} to O_{pb} **can be very expensive.** But moving from O_{pb} to O_{ph} is **cheap,** utilizing linear combination of unitaries (LCU).

Conversion between function oracles (1)

So we have this state of affair for encoding function and converting between different oracles:

- Moving from O_b (binary) to O_{pb} (probability): we should convert directly, without intermediate conversion to O_{ph} (phase).
- Probability oracle O_{pb} is basically equivalent to analog oracle O_a, since same tools like quantum Fourier transform can be utilized. So whenever possible, we should aim to use probability oracle instead of analog oracle. Moving from O_a to other oracles is heavily infeasible.
- Moving from O_{pb} (probability) to O_b: this turns out to be very difficult. While it is easy to convert from O_{pb} to phase oracle O_{ph}, it is difficult to convert from O_{ph} to O_b. And direct conversion requires amplitude estimation, which depending on function $f(t)$ can be very expensive.

Conversion between function oracles (2)

So we have problems. Different quantum algorithms require different function oracles, and their quantum power is largely not compatible with each other. In other words, they cannot be utilized together. To restate the previous slide more concisely,

- $O_{ph} \to O_b$: difficult. (but $O_b \to O_{ph}$ easy.)
- $O_{pb} \to O_b$: difficult. (but $O_b \to O_{pb}$ easy.)
- $O_{ph} \to O_{pb}$: difficult. (but $O_{pb} \to O_{ph}$ easy.)

So except for conversion $O_b \to O_{pb} \to O_{ph}$, oracle conversions can only be guaranteed to be easy once. Note: these are current states of affair, not hard bounds.

Back to the background story

So when we want to manipulate function $f(t)$ involving quantum Fourier transform (QFT), we could utilize O_b, convert to O_{pb} and apply QFT. But we cannot easily convert back to binary encoding such that we can utilize algorithms like period finding.

This is a sad story that seems to validate the interpretation of the famous BBBV theorem and its generalizations as 'almost' proving that NP is not contained in BQP.

But if we assume as an oracle something like the binary oracle analog of QFT, something like binary-oracle QFT (BOQFT), then we may still efficiently perform some quantum operations...

Back to the background story (2)

So I wondered: wouldn't it make sense to consider what would happen if BOQFT is available such that even if function is represented in binary oracle, we may do a bunch of things? Of course this requires us to place function values in superposition such that sufficient number of samples of $f(t)$ is available.

For sure, people that have heard this line of thoughts responded....

- Something like BOQFT is almost surely going to be non-unitary, so why bother?
- We should not abuse oracles - sometimes they need to be implemented explicitly.
- We expect the spirit of BBBV theorem to generalize, so again, why bother?

But...

OK, so say, BOQFT doesn't make sense. But maybe we can do more careful works with the binary oracle and then convert to the probability oracle for powerful quantum applications?

After all, in most cases, we do not just have access to value of $f(t)$. We can do fair manipulations of $f(t)$ or provide multiple functions that relate to $f(t)$ and their binary oracles.

Furthermore, we still have the route of $O_b \rightarrow O_{pb} \rightarrow O_{ph}$. Maybe we can find a more general use of O_{ph} that allows us to evade the BBBV theorem.

If BBBV continues to generalize

If the BBBV theorem continues the trend of turning out to be fairly general, it implies that the $O_{ph} \rightarrow O_b$, $O_{pb} \rightarrow O_b$ and $O_{ph} \rightarrow O_{pb}$ conversions are guaranteed to remain difficult.

Which, of course, means that quantum computing unfortunately is not that much powerful and can be very difficult to work with. This contrasts with classical computing, where encoding methods are not that much of a problem!

The End

Thank You!