# Robust Quantum Minimum Finding with an application to hypothesis selection

Yihui Quek
(Joint work with Clement Canonne (IBM Research Almaden),
Patrick Rebentrost (CQT-NUS))

Stanford University

21 Apr 2020

# Problem: Hypothesis selection

## Problem (Hypothesis selection)

*Given*

- *Unknown probability distribution $p_0$, sample access to it.*
- *$N$ known candidate distributions: $\mathcal{P} = \{p_1, \ldots p_N\}$; PDF comparator between every pair.*

*Task: Output a distribution $\hat{p} \in \mathcal{P}$ with small $\ell_1$-distance to $p_0$ with as few samples from $p_0$ as possible.*

Remark: Maximum likelihood does not work for $\ell_1$-distance.

# A closely related problem

## Problem (Robust minimum finding)

*Given*

- *A list of N elements* $\{x_i\}_{i=1}^{N}$
- *A well-defined distance metric* $d(x_i, x_j)$

*Task: Find the minimum using an imprecise pairwise comparator between elements.*

- **Comparator imprecision:** Outputs correct answer if the elements are far enough apart; otherwise no guarantees.
- Result: can do this in $\tilde{\mathcal{O}}(\sqrt{N})$ comparator invocations.

# Noisy Comparator

Input to comparator: indices $i$, $j$.

$$\text{NoisyComp}(i,j) = \begin{cases} \operatorname{argmin}\{x_i, x_j\} & \text{if } d(x_i, x_j) > 1 \\ \text{unknown (possibly adversarial)} & \text{otherwise.} \end{cases} \quad (1)$$

## Definition (Oracle notation)

*Will denote oracle implementing noisy comparator as $\hat{O}$ and noiseless comparator as $\hat{O}^{(0)}$.*

**Will count the number of calls of either $\hat{O}$ or $\hat{O}^{(0)}$.**

# Classical noisy minimum selection

**Definition ($t$-approximation)**

*An element $y \in \mathcal{L}$ is a $t$-approximation of the true minimum $y^*$ if it satisfies $d(y, y^*) < t$.*

**Lemma (Optimal approximation guarantee)**

*To get a $t$-approximation for $t < 2$, $P[\mathrm{error}] \geq \frac{1}{2} - \frac{1}{2\mathrm{N}}$.*

Hence, will aim for a 2-approximation guarantee.

# Classical noisy minimum selection, part 2

Run time dependence on $N$? Classically, linear is optimal.

**Theorem ($\mathrm{COMB}$ (Theorem 15 of [AFJOS'16]))**

*A classical randomized algorithm, $\mathrm{COMB}(\delta, S)$, outputs a 2-approximation of the minimum w.p $\geq 1 - \delta$, using $\mathcal{O}\left(N \log \frac{1}{\delta}\right)$ queries to the noisy comparator.*

**We will do this in sublinear – i.e. $\tilde{\mathcal{O}}(\sqrt{N})$ time.**

**Assumption**

*There exists $\Delta \in [N]'$ such that at most $2\Delta$ elements are contained in the fudge zone of any element in the list.*

- Reasonable assumption in most cases, including hypothesis selection.

# Recap: Dürr-Høyer Quantum Minimum Finding

Quantum Minimum Searching Algorithm

1. Choose threshold index $0 \le y \le N-1$ uniformly at random.
2. Repeat the following and interrupt it when the total running time is more than $22.5\sqrt{N} + 1.4\lg^2 N$.[1] Then go to stage 2(2c).
   (a) Initialize the memory as $\sum_j \frac{1}{\sqrt{N}}|j\rangle|y\rangle$. Mark every item $j$ for which $T[j] < T[y]$.
   (b) Apply the quantum exponential searching algorithm of [2].
   (c) Observe the first register: let $y'$ be the outcome. If $T[y'] < T[y]$, then set threshold index $y$ to $y'$.
3. Return $y$.

Figure: Durr-Hoyer '96. Exponential search algorithm = BBHT '98.

Key point: quantum exponential search rapidly moves the pivot to lower ranks.

# Some initial observations

- What happens if we naively run Dürr-Høyer with noisy comparator? Problem: we could in principle go back up the ranks!
- Will show that we still make (on expectation) positive progress down the ranks, if rank of pivot is $\Omega(1 + \Delta)$.
- However, this stops working when pivot is $o(1 + \Delta)$ ranks from the minimum.
- V1 algorithm: stop iterating when pivot is, on expectation, $\leq O(1 + \Delta)$ ranks from the minimum: already an improvement from $O(N)$.

# Subroutine: QSearchWithCutoff

- We add an explicit run time cutoff to exponential search and allow to use noisy oracle, $\hat{O}$.

## Lemma

*Let the current pivot $y$ be of rank $r > \Delta$. Then* $\textsc{QSearchWithCutoff}(\hat{O}, y, 9\sqrt{\frac{N}{r-\Delta}})$ *succeeds in finding a marked element with probability at least $\frac{1}{2}$.*

## Proof.

Follows from running for twice the expected runtime. □

Will present 3 algorithms.

# Algorithm 1: Pivot Counting QMF

# Algorithm 1: Pivot-Counting QMF

Differs from Dürr-Høyer Quantum Minimum Finding in 2 ways:

- At each iteration, we run $\textsc{QSearchWithCutoff}$ *with a finite cutoff* at $9\sqrt{\frac{N}{1+\Delta}}$ – i.e. may fail to change the pivot.
- Instead of cutting off total runtime, we cut off the number of runs of $\textsc{QSearchWithCutoff}$ at $N_{\text{trials}}$ runs.

ill differentiate between 'attempted' vs. 'successful' pivot change.

# PCQMF pseudocode

**Algorithm 1** Pivot-Counting Quantum Minimum Finding PIVOTQMF($\hat{O}$, $\Delta$, $N_{\text{trials}}$)

**Input:** comparison oracle $\hat{O}, \Delta \in [N]', N_{\text{trials}} \in \mathbb{Z}_+$

$k \leftarrow 0$.
$y \leftarrow \text{Unif}[N]$.
**for** $k \in [N_{\text{trials}}]$ **do**
  $(y', 0) \leftarrow \text{QSEARCHWITHCUTOFF}(\hat{O}, y, 9\sqrt{\frac{N}{1+\Delta}})$.
  **if** $O_y(y') = 1$ **then** $y \leftarrow y'$.

**Output:** $y$

For the next few slides, we will pretend all pivot changes are successful.

# Lemma: worst-case expected improvement

- For 'high' rank':

Lemma (Worst-case expected rank change for $r_i > 3(\Delta + 1)$)

*For* $r_i > 3(\Delta + 1)$,

$$\mathbb{E}[r_{i+1} \mid r_i] \leq \frac{1}{r_i + \Delta} \sum_{s=1}^{r_i+\Delta} s = \frac{r_i + \Delta + 1}{2} < \frac{2r_i}{3}.$$

- For 'low' rank:

Lemma

*For* $r_i \leq 3(\Delta + 1)$, $\mathbb{E}[r_{i+1} \mid r_i] \leq 4\Delta + 3$.

# Crucial intuition

With noiseless comparator:

$$\mathbb{E}[r_{i+1} \mid r_i] \leq \frac{r_i}{2}.$$

i.e. length of 'remaining list' halves with every pivot change.

- **With every successful pivot change, we still make positive progress down the ranks with noisy comparator!** – with worse factor: $2/3$.
- Hence number of successful pivot changes necessary still logarithmic-in-$N \sim \log_{3/2}(N)$.

# Completing the argument

Argument so far: with $N_p = \left\lceil \frac{\log\left(\frac{N}{4\Delta+3}\right)}{\log(3/2)} \right\rceil$ successful pivot changes, expected final rank $\leq 4\Delta + 3$.

To finish off:

- Each attempt succeeds with probability $> \frac{1}{2}$. Chernoff bound number of attempts to get $N_p$ successful pivot changes.
- Markov's inequality bounds actual final rank (pay a multiplicative factor).

| Name | Success prob. | Final guarantee | Run time |
|---|---|---|---|
| PivotQMF | $\frac{3}{4}$ | $\text{rank}(y) \leq 16\Delta + 16$ | $\tilde{\mathcal{O}}(\sqrt{\frac{N}{1+\Delta}})$ |

# Algorithm 2: Repeated PivotQMF

# Algorithm 2: Repeated PivotQMF

- Use PivotQMF as a basic subroutine to find some element of "pretty-good" rank with constant probability
- Repeat $\log(1/\delta)$ times to get a pool of indices.
- Use classical min selection on the pool.

---

**Algorithm 2** Repeated Pivot Quantum Minimum Finding RepeatedPivotQMF$(\hat{O}, \delta, \Delta)$

---

**Input:** $\hat{O}, \delta, \Delta$

$S \leftarrow \varnothing$.

**Stage I: Finding pretty-small element w.h.p.**

**for** $i = 1, \dots \log_4(2/\delta)$ **do**

$y \leftarrow \textsc{PivotQMF}\left(\hat{O}, \Delta, \lceil 8 \max(\frac{\log(N/(4\Delta+3)}{\log(3/2)}, 2\ln N)\rceil\right)$

$S \leftarrow S \cup \{y\}$.

**Stage II: Classical minimum selection with noisy comparator**

Perform $\mathrm{COMB}(\delta/2, S)$.

**Output:** Output of $\mathrm{COMB}(\delta/2, S)$.

---

# Guarantees

| Success prob. | Final guarantee | Run time |
|---|---|---|
| $1-\delta$ | $\text{rank}(y) \leq 18\Delta + 16$ | $\tilde{\mathcal{O}}(\sqrt{\frac{N}{1+\Delta}}\log(N) + \log^2(1/\delta))$ |

Intuition:

- Use quick and dirty quantum subroutine to find a ‘representative’ element. Bootstrap with repetitions.
- Use slow and precise classical subroutine to select the best of the repetitions.

# Algorithm 3: RobustQMF

# RobustQMF overview

- Rank approximation guarantee of PivotQMF can be strengthened to a distance guarantee.
- Key idea of RobustQMF:
  - Run PivotQMF, get a "pretty-good element" $Y_{out} \sim O(\Delta + 1)$ rank
  - *Get classical sublist of elements ranking below $Y_{out}$
  - Run classical minimum-selection algorithm.
- Final approximation guarantee: a 2-approximation of the minimum! $\sim$ optimal.

* This *almost* works, but our runtime cutoff for exponential search at $9\sqrt{\frac{N}{1+\Delta}}$ now comes back to bite us...

# Two refinements

- Getting the classical sublist of elements ranking below $Y_{out}$
  - Fix $Y_{out}$ as a pivot, then repeatedly apply $\textsc{QSearchWithCutoff}(\hat{O}, Y_{\text{out}}, 9\sqrt{\frac{N}{1+\Delta}})$.
- The run time cutoff problem: For $Y_{\text{out}}$ of rank $< 1 + 2\Delta$, expected run time of exponential search is $> O(\sqrt{\frac{N}{1+\Delta}})$, hence the above run time cutoff may be premature!
  - Denominator of run time cutoff comes from number of marked elements.
  - Append $K$ dummy elements to the list that will always be marked ($K = 2\Delta$ works).

# RobustQMF

**Algorithm 3** Robust Quantum Minimum Finding $\textsc{RobustQMF}(\hat{O}, \delta, \Delta)$

**Input:** $\hat{O}, \delta$

**Stage I: Finding a "pretty-small" element with RepeatedPivotQMF**

$Y_{\text{out}} \leftarrow \textsc{RepeatedPivotQMF}(\hat{O}, \delta/2)$

**Stage II: Finding even smaller elements**

$S \leftarrow \{Y_{\text{out}}\}$

$\tilde{T} \leftarrow 19\Delta + 16$

**for** $i = 1, \ldots 2 \ln 2 \log(\frac{4}{\delta}) \tilde{T}$ **do**

$(y_k, g) \leftarrow \textsc{QSearchWithCutoff}(\hat{O}, Y_{\text{out}}, 9\sqrt{\frac{N}{1+\Delta}}, \text{list} = \mathcal{L}')$.

**if** $O_{Y_{\text{out}}}(y_k) = 1$, and $y_k$ is not a dummy element **then** $S \leftarrow S \cup \{y_k\}$

Remove repeated elements from $S$.

**Stage III: Classical minimum-selection with noisy comparator**

Perform $\mathrm{COMB}(\delta/4,\ S)$.

**Output:** Output of $\mathrm{COMB}(\delta/4,\ S)$.

# Table of main algorithms

| Name | Success prob. | Final guarantee | Run time |
|---|---|---|---|
| PivotQMF | $\frac{3}{4}$ | $\text{rank}(y) \leq 16\Delta + 16$ | $\tilde{\mathcal{O}}(\sqrt{\frac{N}{1+\Delta}})$ |
| Repeated PivotQMF | $1 - \delta$ | $\text{rank}(y) \leq 18\Delta + 16$ | $\tilde{\mathcal{O}}(\sqrt{\frac{N}{1+\Delta}})$ |
| RobustQMF | $1 - \delta$ | $d(y, y^*) \leq 2$ (optimal) | $\tilde{\mathcal{O}}(\sqrt{N(1+\Delta)})$ |

Table: Comparison of quantum minimum-finding algorithms. $y^*$: true minimum.

# Application: sublinear-time hypothesis selection

# Scheffé test (algorithm)

Unknown distribution $q$ (sample access). Want to choose the closer of two hypotheses, $p_i, p_j$.

---

**Algorithm 4** Scheffé test

---

**Input:**

Access to distributions $p_i \in \mathcal{P}$ and $p_j \in \mathcal{P}$,
$\{x_k\}_{k=1}^{K}$ i.i.d. samples from unknown distribution $q$.

Compute $\mu_{\mathcal{S}} = \frac{1}{K} \sum_{k=1}^{K} 1_{x_k \in \mathcal{S}_{ij}}$.

**Output:** If $|p_i(\mathcal{S}_{ij}) - \mu_{\mathcal{S}}| \leq |p_j(\mathcal{S}_{ij}) - \mu_{\mathcal{S}}|$ output $p_i$, otherwise output $p_j$

---

# Algorithm by picture

For a pair of hypotheses, $p_1, p_2$ ...

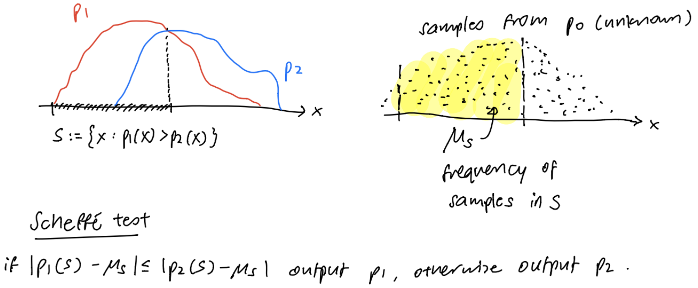


Figure: Scheffe test

# Scheffé test guarantees

W.p. $1 - \delta$, the Scheffé test between two hypotheses outputs an index $i \in [2]$ such that

$$\|p_i - q\|_1 \leq 3 \min_{j \in [2]} \|p_j - q\|_1 + \varepsilon \tag{2}$$

with $\tilde{\mathcal{O}}(\log(1/\delta)/\varepsilon^2)$ samples.

- Scheffé test functions as a **noisy comparator on two hypotheses' $\ell_1$-distance to the unknown distribution** $p_0$.
- Sample-optimal: note no dependence on domain size.

# Map to Robust Quantum Minimum Finding

That is, w.h.p, output of Scheffé test satisfies

$$\begin{cases} p_i & \text{if } \|p_i - q\|_1 < \frac{1}{3}\|p_j - q\|_1 \\ p_j & \text{if } \|p_j - q\|_1 < \frac{1}{3}\|p_i - q\|_1 \\ \text{either } p_i \text{ or } p_j & \text{otherwise.} \end{cases} \tag{3}$$

Taking $x_i = -\log_3 \|p_i - p_0\|_1$, reduces to familiar noisy comparator.

$$\boxed{\mathrm{NComp}(i,j) = \begin{cases} \text{argmin}\{x_i, x_j\} & \text{if } d(x_i, x_j) > 1 \\ \text{unknown (possibly adversarial)} & \text{otherwise.} \end{cases}} \tag{4}$$

# Theorem for sublinear time hypothesis selection

## Theorem (Robust QMF for hypothesis selection)

*Given Assumption 1, there exists a quantum algorithm with expected number of oracle queries*

$$\mathcal{O}\left(\sqrt{\frac{N}{1+\Delta}}\tilde{R}\log\left(\frac{1}{\delta}\right)+(1+\Delta)\log\left(\frac{1}{\delta}\right)\right),$$

*where* $\tilde{R} := \max\left(\mathcal{O}\left(\log\left(N\right)\right),\mathcal{O}\left(\Delta+1\right)\right)$, *that with probability at least* $1-\delta$ *outputs a hypothesis* $\hat{p}$ *that satisfies*

$$\|\hat{p}-q\|_1 \leq 9\min_{p\in\mathcal{P}_\delta}\|p-q\|_1 + 4\sqrt{\frac{10\log\frac{\binom{N}{2}}{\delta}}{k}}. \tag{5}$$

# Concluding remarks

- We have exhibited an algorithm that preserves the quadratic speedup of Durr-Hoyer even with a noisy comparator and obtains optimal approximation guarantees.
- Can do hypothesis selection on $N$ hypotheses in sublinear-in-$N$ time with optimal sample complexity.
- Take-home message:
  - Use a quick and dirty quantum subroutine to reduce 'problem size' (here, find a pretty-good element); finish off with a slow and precise classical subroutine.
- Open questions:
  - In which other contexts is this comparator model valid?
  - Can we do without the assumption that the fudge zone is upper-bounded by $\Delta$?