# Note on the Necessity of Correlation in PCP Queries

Ann He

2020

## Introduction

This paper is concerned with the following question: are there non-trivial PCPs [Aro+98] where all verifier queries are independent?

In this paper, we formally prove the intuition that correlated queries are essentially for the soundness of PCPs. More specifically, we are concerned with PCPs for NP. NP as a class captures the notion of proof verification as polynomial-time efficient. It is usually implied that the verifier can read the entire witness. However, when the verification is restricted to reading only a constant number of locations on a proof, it is intuitive that these locations must be "carefully selected," with knowledge of the proof's encoding. For instance, in the Hadamard PCP [Fal11], the verifier queries three random positions with the third position being the XOR of the first two.

The rest of the manuscript shows that allowing for constant query PCPs, where queries are made uniformly at random and independently, for NP on constant or polynomial-size alphabets, collapses NP to RP, indicating that such PCPs are highly unlikely. We then extend the proof to the case of indpendent queries, answering the original question. [1]

## Constant-Size Alphabet and Uniformly Random Queries

If the PCP verifier makes uniformly random queries, the set of possible PCP proofs can be compressed to a set of histograms encoding of the number of times each alphabet symbol appears in a proof. Therefore any PCP with sub-linear-size alphabets cannot support uniformly random queries unless the exponential-time hypothesis is false.

The subexponential time algorithm queries a histogram encoding the relative frequency of symbols in all possible PCP proofs to compute $max_\pi[A_{\pi,x}]$ where $A_{\pi,x}$ is the probability that the verfier would be convinced that $x \in L$ when it makes random queries to $\pi$. (That is, the the RP algorithm iterates over all possible histograms, querying as the PCP verifier does on each histogram, and accepting iff at least one run of the PCP verifier accepts.)

---

[1] Of course, if one degenerately allows the query complexity to grow arbitrarily, then uniformly querying the PCP proof recovers the original proof, so we restrict our examination to constant query PCPs. Similarly, if one allows for exponentially-sized alphabets, then we can view each alphabet symbol as an encoding of a unique PCP proof, and a uniform-query PCP results from querying at the only index available.

## Polynomial-Size Alphabet and Uniformly Random Queries

In the constant alphabet case, each original proof can be mapped to a histogram with no loss of information relative to the PCP verifier which makes uniformly random and independent queries. An RP algorithm then decides the NP language by trying all possible proofs for an instance $x$. In the polynomial-size alphabet case, a similar approach to achieving an RP algorithm requires us to cover an exponentially sized set of proof strings by a polynomially sized set of proof string "summaries." We propose to perform the "covering" as follows–instead of checking all possible histograms of symbol frequency, we compute a random but constant-length summary of length $100k^2$ of the PCP proof for each $x$. This process is formalized by a modified PCP using these short, random proof summaries. The soundness of the PCP comes from the following averaging argument [Bar06]–if the verifier accepts with good probability on a random histogram, then there exists a fixed histogram for which the verifier accepts with good probability. Below, we formally give the theorem and its proof.

**Definition 0.1** (PCP for NP)**.** *Let $\langle P, V\rangle$ be a PCP for NP using a polynomial-size alphabet, making only a constant number $k$ of uniformly random queries and has perfect completeness and soundness $s$ where $0 < s < 1$ is a constant. That is for any $x \notin L$, and for all possible proof strings $\pi$, the probability that $V$ accepts on $\pi$ and $x$ (where the probability is over $V$'s coins) is less than $s$.*

**Theorem 0.2** (Main Theorem)**.** *Let $PCP = \langle P, V\rangle$ be any PCP system for an NP-complete language. Unless $RP = NP$, the queries $V$ make cannot be independent and uniformly at random.*

**Algorithm 1** PCP for NP with Sampling

```
inputs: $\langle P, V\rangle$, $x$
run $P(x)$ to obtain $\pi$
sample $100k^2$ locations independently and uniformly at random from $\pi$ to create string $b$
run $V$ on $x$
answer $V$'s queries using $b$
output whatever bit $V$ outputs
```

*Proof.* Let $\langle P^*, V^*\rangle$ denote the PCP with sampling to distinguish it from a vanilla PCP. The completeness of $\langle P^*, V^*\rangle$ is self-evident–by the perfect completeness of the PCP protocol, for $x \in L$ any $k$-tuple sample of symbols which appears in the original proof will lead to the verifier accepting. To argue the soundness of $\langle P^*, V^*\rangle$, we argue that for $x \in L$, any proof summary which leads to verifier acceptance is not far in statistical distance from a proper PCP proof which leads to verifier acceptance. The intuition is that the if there are no collisions in the the sampling process from the intermediate string to the final $k$-tuple that the verifier sees, then the new sampling strategy essentially replicates sampling uniformly at random from the original proof string.

### Bounding the soundess of $\langle P^*, V^*\rangle$

**Lemma 0.3** (Distinguishing Distance)**.** *Let $R(X)$ be a random variable denoting $V$'s output when sampling is done according to $\langle P^*, V^*\rangle$ and let $R(Y)$ be one denoting $V$'s output when sampling is done according to $\langle P, V\rangle$. Then $|Pr(R(X) = 1) - Pr(R(Y) = 1)| < c \cdot s$ where $0 < c < 1$ is a constant.*

Let $C$ be the event of collision. Specifically, let $C$ be the event that there is a collision in the sampling process from $b_\pi$ to $a = (a_1, ..., a_k)$, i.e. that the verifier samples the same index in $b$ more than once. More formally, let $C$ be the event that $a_i = b_j = a_{i'}$ for $i \neq i'$. Then using the law of total probability,

$$\begin{aligned}|Pr(R(X) = 1) - Pr(R(Y) = 1)| = |Pr(R(X) = 1|\neg C)(1 - Pr(C)) +& \\ Pr(R(X) = 1|C)Pr(C) - Pr(R(Y) = 1)|&\end{aligned}$$

Replacing $Pr(R(X) = 1|\neg C)$ with $Pr(R(Y) = 1)$ and collecting like terms, we have

$$\begin{aligned}&= |Pr(R(X) = 1|C)Pr(C) - Pr(R(Y) = 1)Pr(C)| \\ &= Pr(C)|Pr(R(X) = 1|C) - Pr(R(Y) = 1)| \\ &\leq s \cdot Pr(C)\blacksquare\end{aligned}$$

In particular, $Pr(R(X) = 1|C) \leq s$ or else there exists a $\pi$ such that $V^\pi(x) > s$ in the original PCP protocol, contradicting its soundness. And $Pr(R(Y) = 1) \leq s$ by soundness of $\langle P, V\rangle$, giving us the final inequality

This implies that for $x \notin L$, the probability that $V^*$ accepts is $\leq s + s \cdot Pr(C)$. If $Pr(C) < 1 - s$, then $s + s \cdot Pr(C)$ is a constant less than 1. □

Using the birthday paradox approximation, we can bound the probability of collision. Specifically, $p(n, d) \approx 1 - e^{-n^2/2d}$ where $p(n, d)$ approximates the probability of throwing two balls into the same bin when throwing $n$ balls into $d$ bins.

$$\begin{aligned}Pr(C) &= 1 - e^{-k^2/200k^2} \\ &= 0.005\end{aligned}$$

For $x \notin \mathcal{L}$, then, the probability that $V^*$ accepts is $< s + 0.0025$.

## RP Protocol

We show that a $PCP$ for $NP$ making a constant number of uniformly random and independent queries collapses $NP$ to $RP$ by explicitly exhibiting an $RP$ machine deciding any $L \in NP$. Let $\mathcal{S}$ denote all the possible histograms of length $100k^2$, and let $N = |\Sigma|$.

**Algorithm 2** RP Protocol $M$

```
Inputs: $x$, $V^*$
y = 0
for $b$ in S do
    set c = 0
    for $i \in \{1, \dots, N\}$ do
        run $V^*$ on $x$, answering any queries it makes according to $b$
        if $V^*$'s output is 1, then do $c = c + 1$
    end for
    if $c == N$, then do $y = y + 1$
end for
if $y > 0$ output 1, otherwise output 0
```

On input $x$, $M$ will iterate through all $|\Sigma|^{100k^2}$ possibilities of the string $b$. For each $b$, $M$ will run $V^*$ a polynomial $N$ number of times on $b$ and $x$. If all $N = |\Sigma|$ iterations of $V^*$ on $b$ and $x$ accept, then $M$ writes down a 1 for that $b$. Otherwise, $M$ writes down a 0. After all $|\Sigma|^{100k^2}$ possibilities $b$ are tried, $M$ accepts if it wrote down at least a single 1. Otherwise, $M$ rejects.

*Proof.* We claim that $M$ is an $RP$ machine deciding $L$.

Let $X_i$ be a random variable representing the symbol $M$ writes down for $b_i$.

Soundness: For $x \notin L$, then $Pr(X_i) = 1 < (s + s \cdot c)^N < \frac{201}{400}^N$. Union bounding over all $N^{100k^2}$ iterations, we have $Pr[M(x) = 1] = \frac{201}{400}^N \cdot N^{100k^2}$, which tends to zero for large $N$, since $k$ is a constant. In order for $M$ to be an $RP$ machine, we need that for all inputs $x \notin \mathcal{L}$, $Pr[M(x) = 1] \leq \frac{1}{2}$, that is we need $(\frac{201}{400})^N \cdot N^{100k^2} \leq \frac{1}{2}$ for all polynomially-bounded $N$.

Taking log to be base 2, we see that the inequality holds when $0.993\sqrt{N} - \frac{1}{\log N} < 100k^2$. As long as $N \geq 2$, $\frac{1}{\log N}$ is between 0 and 1, so

$$0.986N < (100k^2 + 1)$$

In other words, $N = |\Sigma|$ is bounded by a constant, so we can run the RP algorithm for the constant-size alphabet instead for the small alphabet case, and reserve the more complicated RP algorithm for the large $N$ case.

Completeness: In the case of $x \in L$, we claim that $Pr[M(x) = 1] = 1$. To see this, let $\pi$ be the proof string that $P$, the PCP prover in the original protocol sends to $V$. Then, for any $b$ which contains only symbols found in $\pi$, $V^*(x) = 1$–otherwise, we could contradict the perfect completeness of the original PCP protocol. □

# Non-uniform but Independent Queries

A natural generalization of the original investigation is to the case of queries which are not not necessarily made uniformly at random, but at least retain the property of $k$-wise independence. The independence property allows us to re-use the useful heuristic of a proof histogram. Because the PCP queries are independent, the probability that the verifier queries a particular proof index is agnostic of the particular proof $\pi$. That is, the query strategy of the verifier can be summarized by a single probability distribution $p(\cdot)$. Thus, instead of checking all possible histograms of proof symbols, the subexponential time algorithm must check all possible "weighted histograms."

**Definition 0.4** (PCP for NP with k-wise independent queries)**.** *Let $\langle P, V\rangle$ be a PCP for NP using a polynomial-size alphabet, making only a constant number $k$ of independent queries and has perfect completeness and soundness $s$ where $0 < s < 1$ is a constant. That is for any $x \notin L$, and for all possible proof strings $\pi$, the probability that $V$ accepts on $\pi$ and $x$ (where the probability is over $V$'s coins) is less than $s$.*

Let $\langle P, V\rangle$ be a PCP for NP with k-wise independent queries. Let $p_V(\cdot)$ denote $V$'s sampling strategy. Fix some arbitrary proof string $\pi$. Let $h_\pi$ denote a weighted histogram of $\pi$. Then sampling $k$ spots from $\pi$ independently according to $p_V(\cdot)$ is the same as sampling $k$ spots from $h_\pi$ uniformly at random and independently.

In order to turn this into an $RP$ protocol for deciding $NP$, we will argue the statistical similarity between four distributions.

- Let $D_0$ be the distribution resulting from sampling according to $p_V(\cdot)$ $k$ independent times, i.e., the distribution seen by $V$.
- Let $D_1$ be the distribution resulting from sampling $h_\pi$ uniformly at random and independently $k$ times.
- Let $D_2$ be the distribution resulting from sampling $h_\pi$ uniformly at random and independently $100k^2$ times to create an intermediate string $b$, and then sampling $b$ uniformly at random and independently $k$ times.
- Let $D_3$ be the distribution resulting from sampling $\pi$ according to $p_V(\cdot)$ $100k^2$ times to create an intermediate string $b$, and then sampling $b$ uniformly at random and independently $k$ times.

**Lemma 0.5** (Distinguishing Distance for Independent Queries)**.** *Let $V$ be a PCP verifier as in 0.4. Let $R(D_0)$ be a random variable denoting $V^*$'s output when its queries are answered according to distribution $D_0$ and let $R(D_3)$ be one when queries are answered according to $D_3$. Then $|Pr(R(D_0) = 1) - Pr(R(D_3) = 1)| < c \cdot s$ where $0 < c < 1$ is a constant.*

*Proof.* To see that $D_0$ and $D_1$ are the same distribution recall that $h_\pi$ is a string encoding $p_V(i)$ proportion of the symbol at $\pi[i]$ for every index $i$ in $\pi$. To bound the distance between $D_1$ and $D_2$, we observe that we can simply apply Lemma 0.3. Distributions $D_2$ and $D_3$ are the same. Applying the triangle inequality to the hybrid, we have

$$\begin{aligned} |Pr(R(D_0) = 1) - Pr(R(D_3) = 1)| &\leq \sum_{i=0}^{2} |Pr(D_i) - Pr(D_{i+1})| \\ &\leq c \cdot s \end{aligned}$$

□

We now prove the Dependent Queries theorem.

**Theorem 0.6** (Dependent Queries)**.** *Let PCP = $\langle P, V\rangle$ be any PCP system for an NP-complete language. Unless RP = NP, the queries $V$ make cannot be independent.*

*Proof.* We observe that $D_3$ can be implemented by re-running $V$ using independent randomness, as $V$'s queries are importantly independent. Then the proof for 0.6 follows the same structure as that of 0.2. In Algorithm 1, instead of directly $100k^2$ locations independently and uniformly at random from $\pi$, we run $V$ $100k$ times to obtain the $100k^2$ independent samples. The RP protocol follows the same format. □

## Acknowledgements

This manuscript is the result of questions the author had while first learning about probabilistically checkable proof systems. The author thanks Justin Holmgren for answering the primary question on uniformly random queries in this work and for coming up with the proof of the main theorem. The author thanks Fermi Ma greatly for helpful feedback and guidance on earlier drafts, and Orr Paradise for suggesting that author attempt the extension to independent queries, as well as Vincent Paul Su, Alex Lombardi, and Li-Yang Tan for helpful discussions.

## References

[Aro+98] S. Arora et al. “Proof verification and the hardness of approximation problems”. In: *Journal of the ACM* 45.3 (1998), pp. 501–555.

[Bar06] Barak, Boaz. *Note on the averaging and hybrid arguments and prediction vs. distinguishing.* 2006.

[Fal11] Falcon, J. and Jain, M. *An Introduction to Probabilistically Checkable Proofs and the PCP Theorem.* 2011.