Approximating Large Matrix A

1. **Importance and Motivation.** Using SVD we can get the approximation of A in the following way.

Let A be our matrix, defined as a set of column vectors. Establish G as a set of Gaussian random row vectors. Define their product Y.

$$
\begin{aligned}
Y &= AG \\
&= \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix}
\end{aligned}
$$

Now derive Q from QR decomposition, $Q =$ `qr(Y, 0)`. We can define $B = Q^*A$, with decomposition $B = \hat{U}DV^*$, and we claim that

$$
\begin{aligned}
QQ^*A &\approx A \\
QB &\approx A \\
Q - \texttt{svd(B)} &\approx A
\end{aligned}
$$

This approximation of A can be widely used in a ton of applications, *not just SVD*! Therefore it's important to understand its derivation and usefulness.

2. **Properties and Definition of Q.** We have shown that $QQ^*A \approx A$, and we can examine the properties of QQ^*. We claim that QQ^* is actually a projector onto the range of Q.

To be a projector, the following properties must be satisfied.

1. $\hat{P}^2 = \hat{P}$
2. $\hat{P}\vec{v} = \vec{v}$ for some vector $\vec{v}$.

These can be trivially shown using the definition of $\hat{P}$, which implies the following two statements are equivalent.

- range$(A) \subseteq$ range(Q)
- $A = QQ^*A$.

In a sense, Q *captures* the range of A.

We can see that these properties hold as long as $Q\underbrace{Q^*A}_{B} \approx A$ *and* B is smaller than A.

3. **Determining Q.** We've defined Q to be the result of QR decomposition on our Y matrix (our random sampling of A), but we have to be careful how we define that random sampling.

$$
\begin{array}{ccc} Y & = & \underset{m \times n}{A} \quad \underset{n \times (k+p)}{G} \end{array}
$$

Much of the algorithm is now determined by $k + p$.

- If $k + p$ is small, A isn't sampled enough and our approximation is no longer accurate.
- If $k + p$ is large, B grows large and we've lost the reason why we're trying to approximate A in the first place.

So how do we determine Q?

4. **Adaptive Range Finder.** [1]

Given $m \times n$ matrix A, a tolerance ϵ, and an integer r, find Q such that

$$\|(I - QQ^*)\, A\| \leq \epsilon$$

holds with probability at least $1 - \min\{m, n\}\, 10^{-r}$.

```
Data: A, ϵ, r
Result: Q
Draw standard Gaussian vectors w^(1), ..., w^(r) of length n.
for i = 1, 2, ..., r do
  | y^(i) = A w^(i)
end
j = 0
Q^(0) = [];                                   /* the m×0 empty matrix */
while max { ||y^(j+1)||, ||y^(j+2)||, ... ||y^(j+r)|| } > ϵ/10 sqrt(2/π) do
  | j += 1
  | Overwrite y^(j) = (I − Q^(j−1) (Q^(j−1))^*) y^(j)
  | q^(j) = y^(j) / ||y^(j)||
  | Q^(j) = [Q^(j−1) q^(j)]
  | Draw a standard Gaussian Vector w^(j+r) of length n.
  | y^(j+r) = (I − Q^(j−1) (Q^(j−1))^*) A w^(j+r)
  | for i = (j + 1) : (j + r − 1) do
  |   | Overwrite y^(i) = y^(i) − q^(j) ⟨q^(j), y^(i)⟩
  | end
end
Q = Q^(j)
```

[1] Algorithm 4.2 in Halko, Martinsson, Tropp, Page 25, `http://arxiv.org/pdf/0909.4061.pdf`