Cheap Checking for Cloud Computing

Statistical Analysis via Annotated Data Streams

Graham Cormode, Christopher Hickey
University of Warwick

Motivation

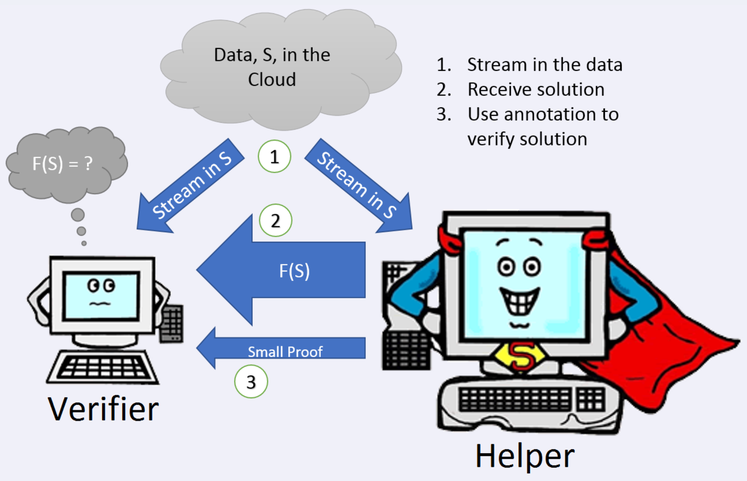
Cloud computing is now an effective way to outsource data analysis.

The issue of trust arises, can the user check a result is optimal without repeating the computation?

How can we verify outsourced computation with limited resources?

Our work focuses dimensionality reduction methods: OLS, PCA and LDA.

Annotated Data Streams



The annotated streaming model [1] works as follows;

- The (weak) Verifier streams the data and performs some preliminary computation
- The (powerful) Helper computes an answer, which is provided along with some additional 'annotation' prescribed by the protocol
- The verifier uses the annotation to help check the result, and either accepts or rejects.

We are looking for protocols that ensure incorrect answers are rejected with high probability, whilst keeping the following costs low

- V - the verifier's memory
- H - the size of the annotation

Fingerprints

Our protocols rely on fingerprints of matrices [3]. These have the property that if two matrix fingerprints agree, then with high probability the matrices are the same.

For $A \in \mathbb{F}_q^{n\times m}$, the matrix fingerprint of A is a linear function with $x \in_R \mathbb{F}_q$

$$f_x(A) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} A_{ij}x^{in+j}$$

We can easily form the fingerprint of a matrix (or vector) as we stream it.

Primitive 1: Matrix Product

Given $A, B \in \mathbb{F}_q^{n\times n}$, we give a protocol where

- $V = O(\log(q))$
- $H = O(n^2\log(q))$

This protocol follows from fingerprinting the outer product of vectors $u, v \in \mathbb{F}^n$

$$f_x(u \otimes v) = f_{x^n}(u)f_x(v)$$

And so

$$f_x(AB) = \sum_{i=0}^{n-1} f_x(A_i^{\downarrow} \otimes B_i^{\rightarrow})$$

$$= \sum_{i=0}^{n-1} f_{x^n}(A_i^{\downarrow})f_x(B_i^{\rightarrow})$$

where $A_i^{\downarrow}$ and $B_i^{\rightarrow}$ are the columns of A and rows of B respectively.

The protocol has the helper replay the input, allowing us to form $f_x(AB)$, and use fingerprints to ensure they match the result.

Our protocol is an improvement by a factor of n over [2].

Primitive 2: Eigenpairs

When verifying eigenpairs of a symmetric matrix $A \in \mathbb{F}_q^{n\times n}$, we must round our solution into a finite field, consequently the solution $(\tilde{v}_i, \tilde{\lambda}_i)_{i=1}^n$ for $A \in \mathbb{F}_q^{n\times n}$ may only approximately satisfy $Av - \lambda v = 0$.

We define a protocol to tolerate approximate solutions $\tilde{\lambda}$ so that $|\lambda - \tilde{\lambda}| \leq \epsilon$, for a parameter $\epsilon > 0$ that can be set arbitrarily small. To reach this precision, we must scale up the field size by a factor T.

Consider $\tilde{v}_i$ and $\tilde{\lambda}_i$, which are $(Tv_i, T\lambda_i)$ rounded into the field $\mathbb{F}_{Tq}$.

$$||TA\tilde{v}_i - \tilde{\lambda}_i\tilde{v}_i||_{\max} \leq O(n||A||_F)$$

This tells us that

$$|T\lambda_i - \tilde{\lambda}_i| \leq O(n^{\frac{3}{2}}||A||_F)$$

Therefore, to achieve error ϵ we choose $T = O(n^{\frac{3}{2}}||A||_F/\epsilon)$, and receive matrices $\tilde{V}, \tilde{D} \in \mathbb{F}_{Tq}$ from the helper, verifying bounds on

$$||TAV - \tilde{D}\tilde{V}||_{\max} \text{ and } ||\tilde{V}^T\tilde{V} - T^2I||_{\max}$$

Our costs are therefore

- $V = O(\log(qT))$
- $H = O(n^2\log(qT))$

Application: PCA

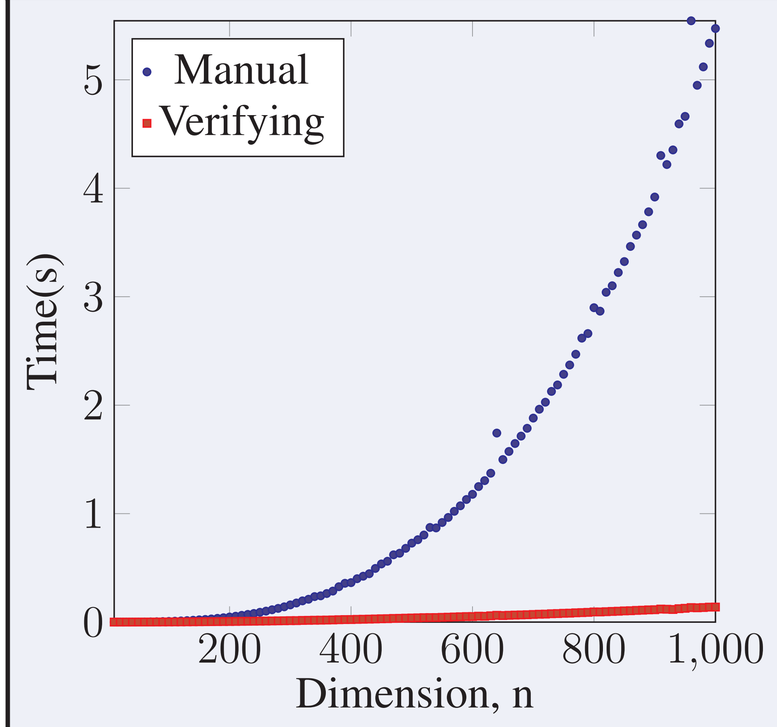
Given a large normalized data matrix $S \in \mathbb{F}_q^{n\times d}$, to find the principal components we must find the eigenvectors of the covariance matrix of the data, S^TS.

We can fingerprint S^TS whilst streaming, and then, invoking our previous primatives, we get a protocol for PCA with precision $\epsilon > 0$ and

- $V = O(\log(qT))$
- $H = O(d^2\log(qT))$

Using $T = O(n^{\frac{3}{2}}||S||_F^2/\epsilon)$.

Practical Results



- Eigendecomposition on randomly chosen $n \times n$ matrices
- Field size $= 2^{31} - 1$, $\epsilon = 0.01$
- Exact computation scales as n^3, while verification time is linear in matrix size
- Memory cost for verifier is negligible (few bytes)
- Bottleneck for Verifier is receiving the eigenvectors (linear in matrix size), a few megabytes in this example

References

References

[1] Amit Chakrabarti, Graham Cormode, and Andrew Mcgregor. Annotations in data streams. *Automata, Languages and Programming*, pages 222–234, 2009.

[2] Samira Daruki, Justin Thaler, and Suresh Venkatasubramanian. Streaming verification in data analysis. In *International Symposium on Algorithms and Computation*, pages 715–726. Springer, 2015.

[3] Michael O Rabin et al. *Fingerprinting by random polynomials*. Center for Research in Computing Techn., Aiken Computation Laboratory, Univ., 1981.