Total Least Squares Regression in Input Sparsity Time

Huaian Diao*, Zhao Song‡, David P. Woodruff†, Xin Yang‡

*Northeast Normal University, †Carnegie Mellon University, ‡University of Washington

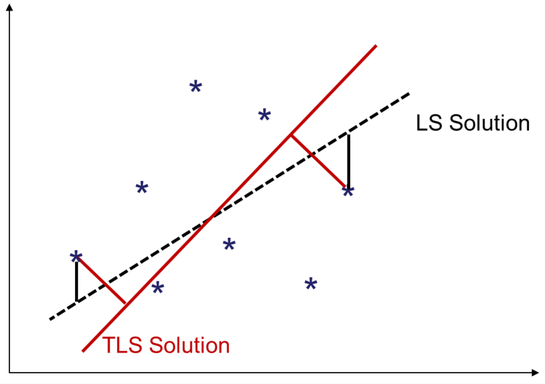
Total Least Squares Problem

Given: $\boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{B} \in \mathbb{R}^{m\times d}$.

Task: Perturb $\boldsymbol{A}, \boldsymbol{B}$ at minimal cost so that the linear system $\boldsymbol{AX} = \boldsymbol{B}$ has a solution.

- Output $\boldsymbol{X} \in \mathbb{R}^{n\times d}$.
- Minimize $||\boldsymbol{A} - \widehat{\boldsymbol{A}}||_F^2 + ||\widehat{\boldsymbol{A}}\boldsymbol{X} - \boldsymbol{B}||_F^2$ over $\widehat{\boldsymbol{A}} \in \mathbb{R}^{m\times d}, \boldsymbol{X} \in \mathbb{R}^{n\times d}$.

Intuition: the ordinary least squares problem $\min_X ||\boldsymbol{AX} - \boldsymbol{B}||_F^2$ assumes only the labels $\boldsymbol{B}$ are corrupted; but the samples $\boldsymbol{A}$ can also be corrupted.



Compact formulation as low rank approximation: let $\boldsymbol{C} = [\boldsymbol{A}, \boldsymbol{B}] \in \mathbb{R}^{m\times(n+d)}$, then the TLS problem is

$$\mathbf{OPT} := \min_{rank(\widehat{C})=n} ||\boldsymbol{C} - \widehat{\boldsymbol{C}}||_F^2$$

Main Results

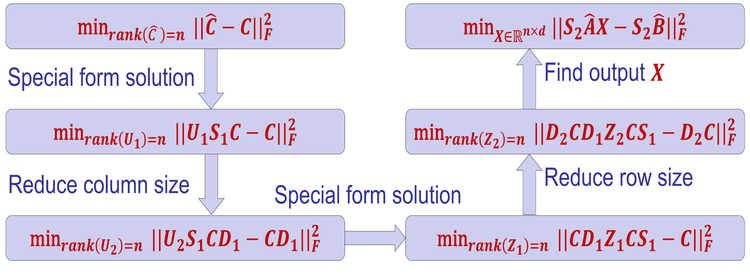
Theorem. For $0 < \epsilon < 1$, with probability 0.9, we can output $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ so that there exists $\widehat{\boldsymbol{A}} \in \mathbb{R}^{m\times n}$ satisfying

$$||[\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{A}}\boldsymbol{X}] - [\boldsymbol{A}, \boldsymbol{B}]||_F^2 \leq (1+\epsilon)\mathbf{OPT}$$

in time $\widetilde{\boldsymbol{O}}(\mathbf{nnz}(\boldsymbol{A}) + \mathbf{nnz}(\boldsymbol{B})) + \boldsymbol{d}\cdot\mathbf{poly}(\boldsymbol{n}/\boldsymbol{\epsilon})$. Here $\mathbf{nnz}$ is the number of non-zero entries, and $\widetilde{\boldsymbol{O}}$ hide some logarithm factors.

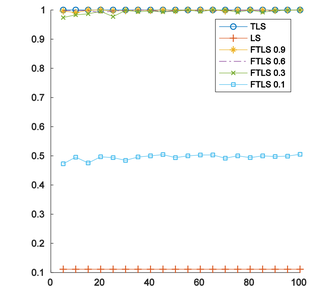
- The output $\boldsymbol{X}$ can be used for future prediction.
- Traditional algorithm for the TLS problem: SVD decomposition, takes $\boldsymbol{O}(\boldsymbol{m}(\boldsymbol{n}+\boldsymbol{d})^2)$ time.
- Our algorithm runs in input sparsity time. Huge advantage when $\boldsymbol{m} \gg \boldsymbol{n}, \boldsymbol{d}$ and the inputs $\boldsymbol{A}, \boldsymbol{B}$ are sparse.
- Our running time is even smaller than writing down the optimal low rank approximation $\widehat{\boldsymbol{C}}$.
- Key ingredient: chain sketching matrices to reduce dimension.

Algorithm Flowchart

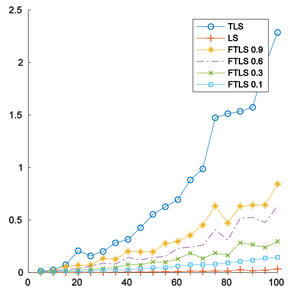


Experimental Results

- Synthetic datasets:



Cost of different algorithms. Measured by algorithm_cost/TLS_cost.



Running time of different algorithms measured in seconds.

- Real datasets: regression datasets from UCI datasets.

Method	Cost	C-std	Time	T-std
TLS	0.10	0	1.12	0.05
LS	10^6	0	0.0012	0.0002
FTLS 0.9	0.10	0.0002	0.16	0.0058
FTLS 0.6	0.10	0.0003	0.081	0.0033
FTLS 0.3	0.10	0.0007	0.046	0.0022
FTLS 0.1	0.10	0.0016	0.034	0.0024

Cost and running time of different algorithms for the Airfoil Self-Noise dataset.

Method	Cost	C-std	Time	T-std
TLS	0.93	0	1.36	0.16
LS	666	0	0.0012	0.001
FTLS 0.9	0.93	0.0032	0.30	0.025
FTLS 0.6	0.94	0.0050	0.17	0.01
FTLS 0.3	0.95	0.01	0.095	0.005
FTLS 0.1	0.99	0.03	0.074	0.004

Cost and running time of different algorithms for the Red wine dataset.

- Our algorithm outperforms the ordinary TLS algorithm by running 10~50x faster.
- Our algorithm obtains small cost comparable to the TLS solution.