Loco: Distributing Ridge Regression with Random Projections

Yang Yu

Department of Statistics
Purdue University

August 16, 2016

(Work by Christina Heinze, Brian McWilliams, Nicolai Meinshausen, and Gabriel Krummenacher)

Outline

- Introduction
  - Goal
  - Related Work
- Problem Setting
  - Feature-wise Distributed Ridge Regression
  - SRHT (Subsampled Randomized Hadamard Transform)
- Algorithm
- Analysis
  - Assumptions
  - Theorem
- Experimental Results
  - Simulated Data
  - Climate Data
- Discussion
  - Summary
  - Future Work
- References

Introduction

Goal

Problem:

- Propose an algorithm for large-scale ridge regression.

Goal:

- Preserve important dependencies between variables.
- Cheap communication and computation.
- Close to the exact ridge regression solution.

Questions:

- How to distribute the data and processing tasks among workers.
- How and what each worker communicate.

Introduction

Related Work

Parallel methods: Parallelize the problem locally amongst multiple cores on the same physical machine with shared memory.

- Cheap communication
- Assuming sparsity
- Impractical for particularly large-scale problems

Distributed methods: Distribute computation amongst K networked workers on a cluster.

- Practical for particularly large-scale problems
- Expensive communication
- Assuming independence between features (Distributing across features)

Introduction

Related Work

Johnson-Lindenstrauss (J-L) projections: A popular method for dimensionality reduction, preserving pairwise ℓ_2 distances between vectors (Ailon and Chazelle, 2009).

- Dimension reduction
- The solution vector is in the compressed space and so interpretability of coefficients is lost.

Problem Setting

Ridge Regression

Ridge regression:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} L(\boldsymbol{\beta}) := n^{-1} \|Y - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \tag{1}$$

- Solution: $\widehat{\boldsymbol{\beta}}^{\text{rr}} = \left(\mathbf{X}^T\mathbf{X} + n\lambda\mathbf{I}\right)^{-1} \mathbf{X}^T Y$
- When p is large, constructing and inverting $\mathbf{X}^T\mathbf{X}$ is prohibitively expensive.
- When n is large, ridge regression is usually solved using stochastic gradient descent (SGD) or stochastic dual coordinate ascent (SDCA) (Shalev-Shwartz and Zhang, 2013).

Problem Setting

Feature-wise Distributed Ridge Regression

Distribute the features across K different workers:

- $\mathcal{P} = \{1, ..., p\}$
- Partition $\mathcal{P}$ into K non-overlapping subsets $\mathcal{P}_1, ..., \mathcal{P}_K$ of equal size $\tau = p/K$
- $\mathcal{P} = \bigcup_{k=1}^{K} \mathcal{P}_k, \quad |\mathcal{P}_1| = ... = |\mathcal{P}_K| = \tau$

A naive attempt: Solving the minimization problem on each subset of features $\mathcal{P}_k$ independently?

- Important dependencies between features in different blocks would not be preserved.

Problem Setting

Feature-wise Distributed Ridge Regression

- $\mathbf{X}_k \in \mathbb{R}^{n \times \tau}$: Columns corresponding to the "raw" features of block k
- $\mathbf{X}_{(-k)} \in \mathbb{R}^{n \times (p-\tau)}$: Remaining columns of $\mathbf{X}$.

Rewrite (1):

$$L(\boldsymbol{\beta}) = n^{-1}\|Y - \mathbf{X}_k\boldsymbol{\beta}_{\text{raw}} - \mathbf{X}_{(-k)}\boldsymbol{\beta}_{(-k)}\|^2 + \lambda\|\boldsymbol{\beta}_{\text{raw}}\|^2 + \lambda\|\boldsymbol{\beta}_{(-k)}\|^2 \quad (2)$$

Idea: Replace $\mathbf{X}_{(-k)}$ in each block with a low-dimensional approximation.

Problem Setting

Feature-wise Distributed Ridge Regression

- $\tilde{\mathbf{X}}_k \in \mathbb{R}^{n \times (K-1)\tau_{\text{subs}}}$: Columns corresponding to the "random" features of block k, a low-dimensional approximation to $\mathbf{X}_{(-k)}$, $\tau_{\text{subs}} \ll \tau$

Define the sub-problem that worker k solves as

$$L_k(\boldsymbol{\beta}_k) = n^{-1}\|Y - \underbrace{\mathbf{X}_k\boldsymbol{\beta}_{\text{raw}}}_{\text{raw}} - \underbrace{\tilde{\mathbf{X}}_k\boldsymbol{\beta}_{k,rp}}_{\text{random}}\|^2 + \lambda\|\boldsymbol{\beta}_{\text{raw}}\|^2 + \lambda\|\boldsymbol{\beta}_{k,rp}\|^2 \quad (3)$$

- $\tilde{\mathbf{X}}_k$: Such that the risk of the minimizer of (3) is similar to the risk of the minimizer of (2) (Use random projections (J-L projections))

Problem Setting

Feature-wise Distributed Ridge Regression

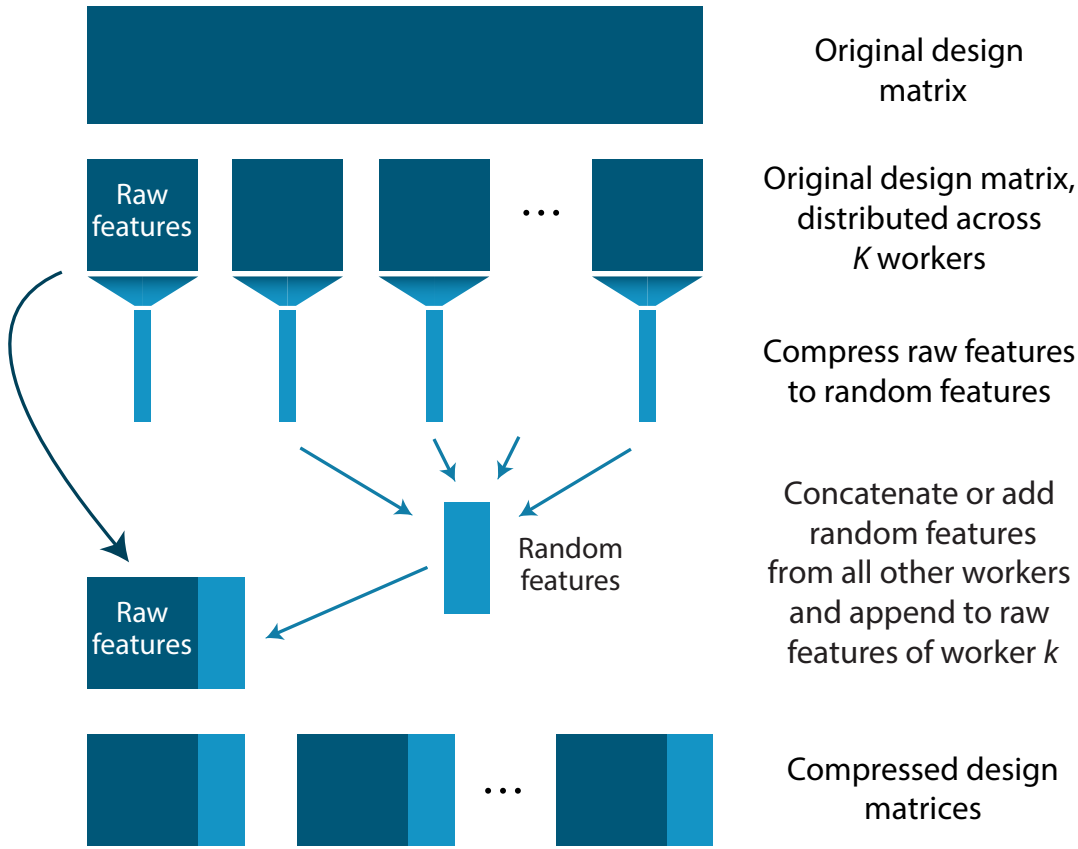


Figure 1: Schematic for the approximation of a large data set in a distributed fashion using random projections. The random features can either be concatenated or added.

Problem Setting

SRHT (Subsampled Randomized Hadamard Transform)

SRHT (Halko et al., 2011; Boutsidis and Gittens, 2012):
$\mathbf{\Pi} \in \mathbb{R}^{\tau \times \tau_{\text{subs}}} = \sqrt{\tau/\tau_{\text{subs}}}\mathbf{DHS}$

- $\mathbf{S} \in \mathbb{R}^{\tau \times \tau_{\text{subs}}}$: A subsampling matrix.
- $\mathbf{D} \in \mathbb{R}^{\tau \times \tau}$: A diagonal matrix whose entries are drawn independently from $\{-1, 1\}$.
- $\mathbf{H} \in \mathbb{R}^{\tau \times \tau}$: A normalized Walsh-Hadamard matrix which is defined recursively as $\mathbf{H} = \frac{1}{\sqrt{\tau}}\mathbf{H}_\tau$,

$$\mathbf{H}_\tau = \begin{bmatrix} \mathbf{H}_{\tau/2} & \mathbf{H}_{\tau/2} \\ \mathbf{H}_{\tau/2} & -\mathbf{H}_{\tau/2} \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}.$$

Problem Setting

SRHT (Subsampled Randomized Hadamard Transform)

SRHT:

- has similar ℓ_2 distance preserving properties as sub-gaussian random projections
- has the added benefit of a fast $O(\tau \log \tau)$ matrix-vector product due to its recursive definition.

Algorithm

Details

Algorithm 1 LOCO

Input: Data: $\mathbf{X}$, Y, Number of blocks: K, Parameters: τ_{subs}, λ

1: Partition $\mathcal{P} = \{1, \ldots, p\}$ into K subsets $\mathcal{P}_1, \ldots, \mathcal{P}_K$ of equal size, τ.
2: **for each** worker $k \in \{1, \ldots K\}$ **in parallel do**
3: Compute and send random projection $\widehat{\mathbf{X}}_k = \mathbf{X}_k \mathbf{\Pi}_k$.
4: Construct $\bar{\mathbf{X}}_k = [\mathbf{X}_k, \tilde{\mathbf{X}}_k]$
5: $\bar{\boldsymbol{\beta}}_k \leftarrow \texttt{SolveRidge}(\bar{\mathbf{X}}_k, Y, \lambda)$
6: $\widehat{\boldsymbol{\beta}}_k = \left[\bar{\boldsymbol{\beta}}_k\right]_{1:\tau}$
7: **end for**

Output: Solution vector: $\widehat{\boldsymbol{\beta}}^{\text{loco}} = \left[\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_K\right]$

Algorithm

Details

Step 3: Each worker computes a random projection, via the SRHT, of its respective block.

- $\widehat{\mathbf{X}}_k = \mathbf{X}_k \mathbf{\Pi}_k \in \mathbb{R}^{n \times \tau_{\textbf{subs}}}$

Step 4: Each worker k constructs the column-wise concatenation of the raw feature matrix $\mathbf{X}_k$ and the random approximations from all other blocks $\tilde{\mathbf{X}}_k$.

- $\bar{\mathbf{X}}_k = \left[\mathbf{X}_k, \tilde{\mathbf{X}}_k\right] \in \mathbb{R}^{n \times (\tau + (K-1)\tau_{\textbf{subs}})}, \quad \tilde{\mathbf{X}}_k = \left[\widehat{\mathbf{X}}_{k'}\right]_{k' \neq k}$

Alternative Step 4: Each worker k constructs the column-wise concatenation of the raw feature matrix $\mathbf{X}_k$ and the sum of the random approximations from all other blocks $\tilde{\mathbf{X}}_k$.

- $\bar{\mathbf{X}}_k = \left[\mathbf{X}_k, \tilde{\mathbf{X}}_k\right] \in \mathbb{R}^{n \times (\tau + \tau_{\textbf{subs}})}, \quad \tilde{\mathbf{X}}_k = \sum_{k' \neq k} \widehat{\mathbf{X}}_{k'}$
- Computed and combined more efficiently

Algorithm

Details

Step 5: The function SolveRidge$(\bar{\mathbf{X}}_k, Y, \lambda)$ returns a vector

- $\bar{\boldsymbol{\beta}}_k = \arg\min_{\boldsymbol{\beta}_k} n^{-1}\|Y - \bar{\mathbf{X}}_k\boldsymbol{\beta}_k\|^2 + \lambda\|\boldsymbol{\beta}_k\|^2 \in \mathbb{R}^{\tau+(K-1)\tau_{\textbf{subs}}}$.

Step 6: The final solution vector $\widehat{\boldsymbol{\beta}}^{\text{loco}}$ is the the concatenation of the first τ coordinates of each $\bar{\boldsymbol{\beta}}_k$ and so lives in the same space as the original data.

- $\widehat{\boldsymbol{\beta}}^{\text{loco}} = \left[\widehat{\boldsymbol{\beta}}_1, ..., \widehat{\boldsymbol{\beta}}_K\right] \in \mathbb{R}^p, \quad \widehat{\boldsymbol{\beta}}_k = \left[\bar{\boldsymbol{\beta}}_k\right]_{1:\tau}$

Algorithm

Computational, Memory and Communication Costs

Memory:

- $\tau + (K-1)\tau_{\mathsf{subs}} \ll p$

Communication:

- One-round

Algorithm

Computational, Memory and Communication Costs

Possible speedups as K increases:

(i) Each local problem becomes easier in a computational sense.
- Computing a random projection in each block: $O(\tau \log \tau_{\text{subs}})$
- Each iteration of the local optimization algorithm: $O(\tau + (K-1)\tau_{\text{subs}})$

(ii) Each local problem becomes easier in a statistical sense (faster convergence).
- Ratio between the number of parameters and the sample size: $(\tau + (K-1)\tau_{\text{subs}})/n$

(iii) As a consequence of (i), the size of the random projections to be communicated by each worker decreases.
- τ_{subs}

Analysis

Notations

Consider the linear model

$$Y = \mathbf{X}\boldsymbol{\beta}^* + \varepsilon, \tag{4}$$

with fixed $\mathbf{X} \in \mathbb{R}^{n \times p}$ and true parameter vector $\boldsymbol{\beta}^* \in \mathbb{R}^p$.

- $\widehat{\boldsymbol{\beta}}^{\text{rr}}$: Ridge estimate for $\boldsymbol{\beta}^*$
- $\widehat{\boldsymbol{\beta}}^{\text{loco}}$: Loco estimate for $\boldsymbol{\beta}^*$
- In the original space, $\widehat{\boldsymbol{\beta}}^{\text{rr}}$ minimizes

$$\min_{\|\boldsymbol{\beta}\| \leq t} n^{-1} \|Y - \mathbf{X}\boldsymbol{\beta}\|^2. \tag{5}$$

- In the compressed space, $\bar{\boldsymbol{\beta}}_k$ minimizes

$$\min_{\|\boldsymbol{\beta}_k\| \leq t} n^{-1} \|Y - \bar{\mathbf{X}}_k \boldsymbol{\beta}_k\|^2. \tag{6}$$

Analysis

Assumptions

Assumption 1

Let $\mathbf{w}^*$ be the true parameter vector after rotating $\mathbf{X}$ to the PCA coordinate system. There exists $1 \leq J \leq \min\{n, p\}$ and $c \in (0, 1/2)$ such that

(A1) the J-th largest eigenvalue of the covariance matrix is strictly positive, that is $\lambda_J > 0$,

(A2) the ridge constraint is active: $t \leq (1 - c) \sum_{j=1}^{J} (\mathbf{w}_j^*)^2$,

(A3) the errors ε_i, $i = 1, \ldots, n$ have zero mean, are independent and their variances are bounded by $\sigma^2 > 0$.

- If (A1) and (A2) do not hold, ridge regression may not be suitable.

Analysis

Risk

Definition 1 (Risk)

Let $\widehat{\mathbf{b}}$ be an estimator for $\boldsymbol{\beta}^*$ and define the risk of $\widehat{\mathbf{b}}$ with fitted values $\widehat{Y} = \mathbf{X}\widehat{\mathbf{b}} \in \mathbb{R}^n$ as

$$R(\mathbf{X}\widehat{\mathbf{b}}) = n^{-1}\mathbb{E}_{\varepsilon}\|\mathbf{X}\boldsymbol{\beta}^* - \mathbf{X}\widehat{\mathbf{b}}\|^2.$$

Analysis

Rate of Convergence

Theorem 1

Under Assumption 1, $\exists\ n_0(\xi)$ for all $\xi > K(\delta + (p-\tau)/e^r)$ such that for all $n \geq n_0$ with probability at least $1-\xi$

$$\mathbb{E}_\varepsilon[\|\widehat{\boldsymbol{\beta}}^{\text{rr}} - \widehat{\boldsymbol{\beta}}^{\text{loco}}\|^2] \leq \frac{5K}{c\lambda_J}\left(\frac{1}{(1-\rho)^2} - 1\right) R(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{rr}})$$

where $\rho = C\sqrt{\frac{r\log(2r/\delta)}{(K-1)\tau_{\text{subs}}}}$, $r = \text{rank}(\mathbf{X})$, λ_J denotes the J^{th} largest non-zero eigenvalue of the covariance matrix and $R(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{rr}})$ is the risk of the ridge estimator. The expectation is conditional on the random projection as the uncertainty coming from the SRHT is captured in the probability with which the statement holds.

- $(1-\rho)^2$: Measures the quality of the random feature representation.
- If K and τ_{subs} are chosen such that $(K-1)\tau_{\text{subs}} \gg r$, the approximation error vanishes.

Experimental Results

Implementation Details

- To guarantee portability across different computing architectures, use a sparse random projection matrix with entries sampled as

$$\mathbf{\Pi}_{i,j} \sim \begin{cases} 1, & \text{w.p. } 1/6 \\ 0, & \text{w.p. } 2/3 \\ -1, & \text{w.p. } 1/6 \end{cases},$$

  which has similar guarantees to the SRHT (Achlioptas, 2003).
- The local ridge regression solver is SDCA (Shalev-Shwartz and Zhang, 2013).
- Use the alternative step 4 (summing random projections) in the algorithm as it allows for a more efficient aggregation of the random projection.
- Compare against CoCoA (Jaggi et al., 2014): A communication efficient approach to dual optimization, distributing across samples

Experimental Results

Simulated Gaussian Data

Two large-scale simulated problems:

- The data is generated from a Gaussian distribution with mean zero and a block-wise covariance matrix such that the features are not independent, which implies that the data is effectively low rank.

Experimental Results

Scenario 1

Scenario 1:

- $n = 4,000, \quad p = 150,000, \quad 600M$ non-zeros, $\quad r = 150$
- $n_{\text{test}} = 1,000$
- $K = 3, 6, 12, 24$
- $\tau_{\text{subs}}/\tau = 0.01, 0.05, 0.1$

Experimental Results

Scenario 1

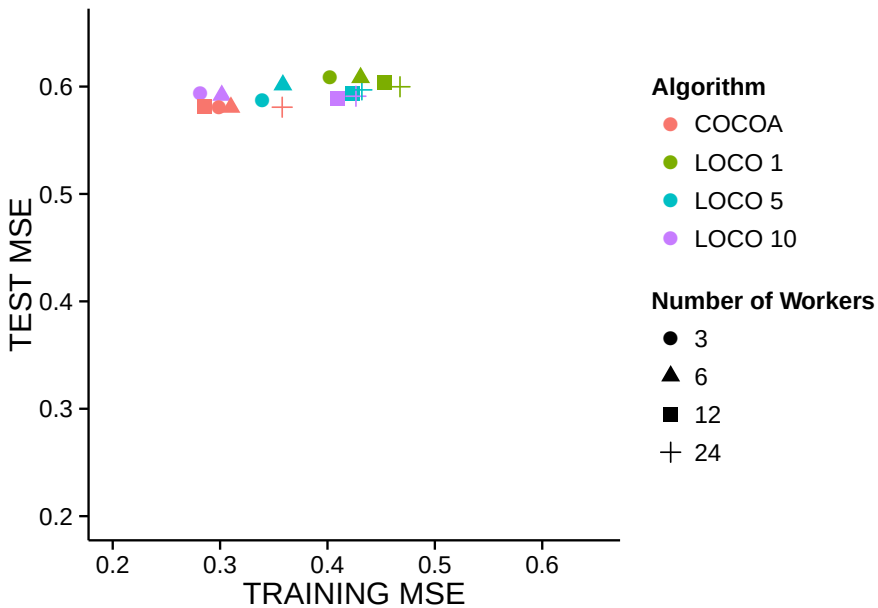
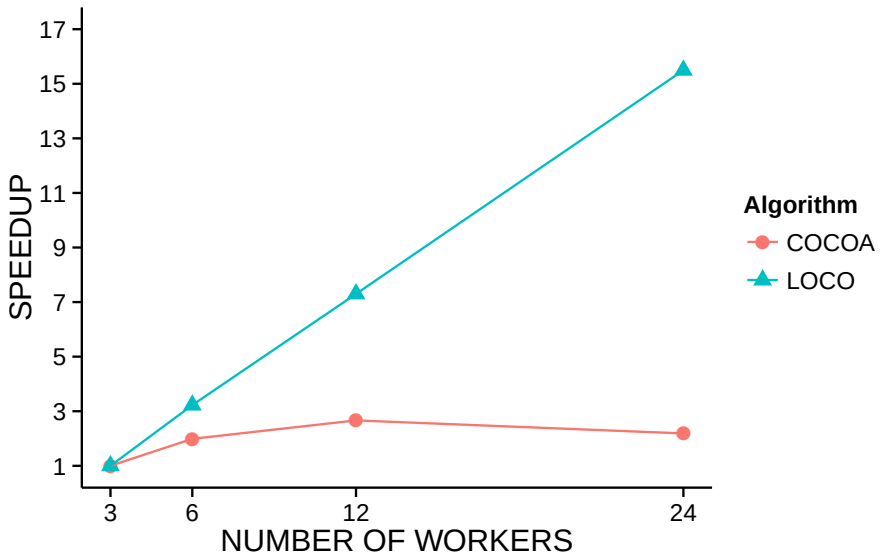


Figure 2: Normalized training and test error when $p = 150,000$.

- Loco 1: $\tau_{\text{subs}}/\tau = 0.01$,...
- As the size of the projection dimension τ_{subs} increases, the performance of Loco improves and approaches that of CoCoA.
- The main difference between CoCoA and Loco lies in the training error. The difference between the test errors are very small.

Experimental Results

Scenario 1



- CoCoA exhibits near-linear speed up for up to 12 workers but a relative slowdown as more workers are added.
- Loco exhibits better-than-linear speedup between 3 and 24 workers.

Figure 3: Relative speedup for different number of workers when $p = 150,000$.

Experimental Results

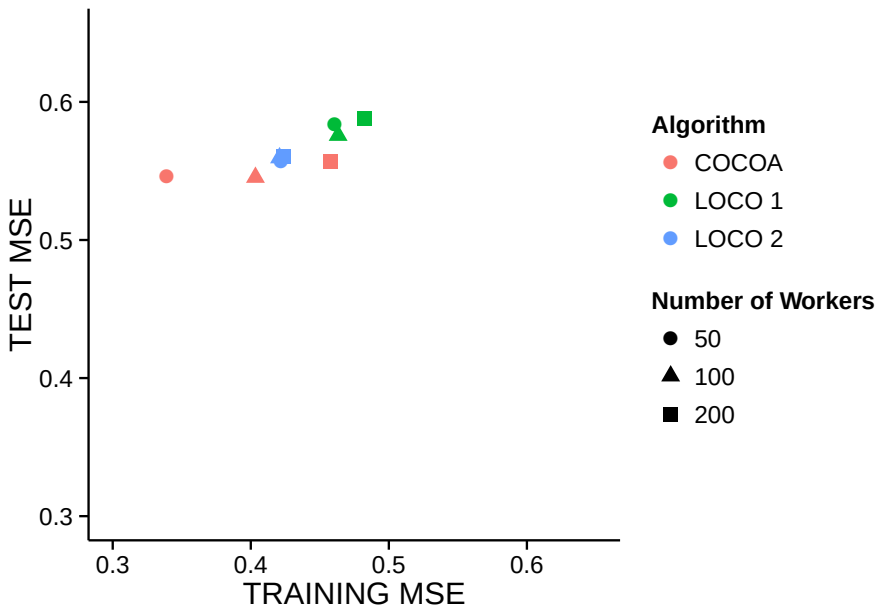
Scenario 2

Scenario 2:

- $n = 8,000, \quad p = 500,000, \quad 4\ billion$ non-zeros, $\quad r = 500$
- $K = 50, 100, 200$
- $\tau_{\text{subs}}/\tau = 0.01, 0.02$

Experimental Results

Scenario 2



- Loco is again able to achieve good test performance, comparable to CoCoA.

Figure 4: Training and test error when $p = 500,000$.

Experimental Results

Scenario 2

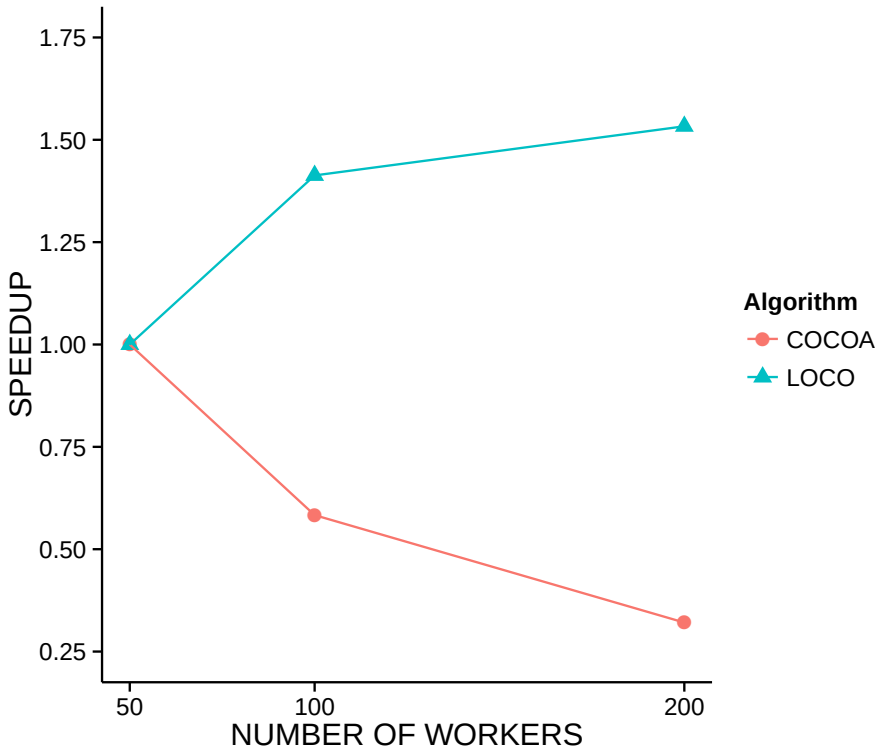


Figure 5: Relative speedup for different number of workers when $p = 500,000$.

When increasing from 50 to 200 workers,

- Loco obtains a $1.5\times$ speedup.
- CoCoA obtains a speedup of 0.32.

Experimental Results

Climate Data

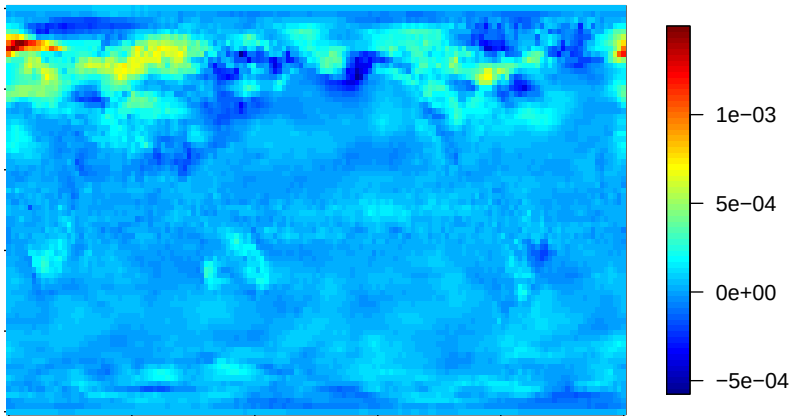
A forecast of the global temperature based on the temperature pattern observed a month earlier.

- Response: the global average temperature in February
- Features: the January temperatures at $10,368$ grid points spread across the globe ($p = 10,368$)
- Data set: $n = 1,062, \quad n_{\text{train}} = 849, \quad n_{\text{test}} = 213$

Ensuring the estimated coefficients are close to the optimal coefficients is in applications like this at least as important as obtaining a low prediction error, since in this application, the regression coefficients have a clear physical interpretation.

Experimental Results

Climate Data



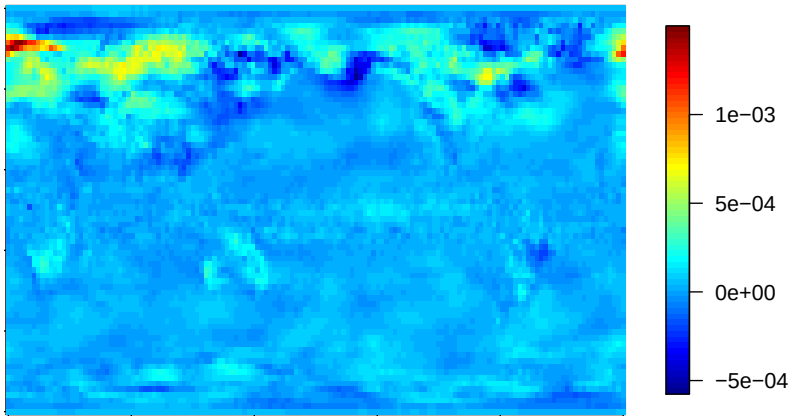
(a) Single machine: Full solution ($MSE = 0.5756$)

Figure 6: Climate data: The regression coefficients are shown as maps with the prime meridian (passing through London) corresponding to the left and right edge of the plot. The Pacific Ocean is occupying the centre of each map.

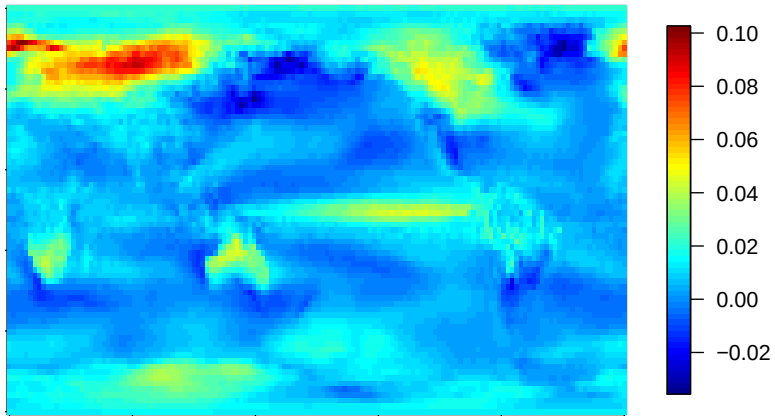
(a) Coefficients estimated in the non-distributed setting with SDCA (near-optimal ridge regression solution on a single machine)

Experimental Results

Climate Data



(a) Single machine: Full solution ($MSE = 0.5756$)



(b) Single machine: Diagonal approximation ($MSE = 53956.4377$)

Figure 7: Climate data: Comparison of different methods which return coefficients lying in the original space.

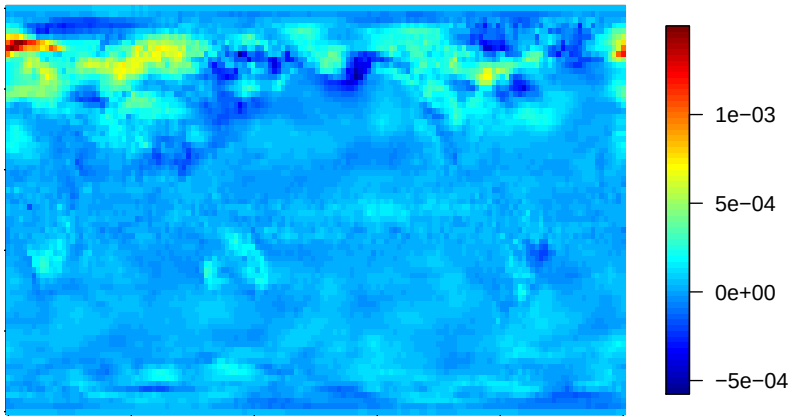
(a) The coefficients estimated in the non-distributed setting with SDCA
(b) The coefficients returned by the naive single-machine approximation

$$\widehat{\boldsymbol{\beta}}^{\text{diag}} = \text{diag}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y, \tag{7}$$

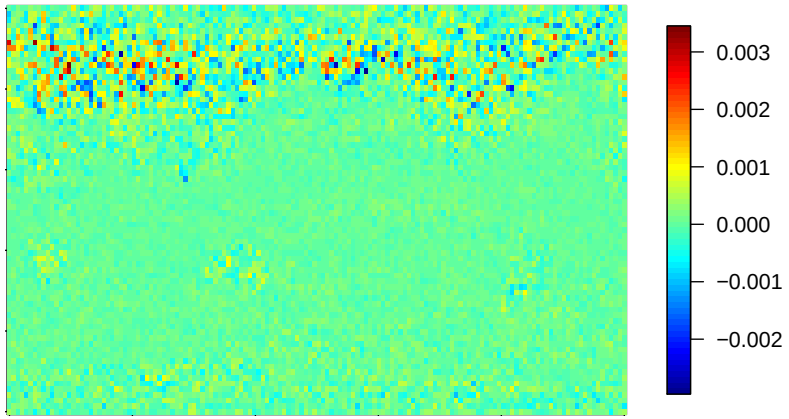
which is equivalent to assuming independence between the features (large MSE, important correlations between features neglected)

Experimental Results

Climate Data



(a) Single machine: Full solution ($MSE = 0.5756$)



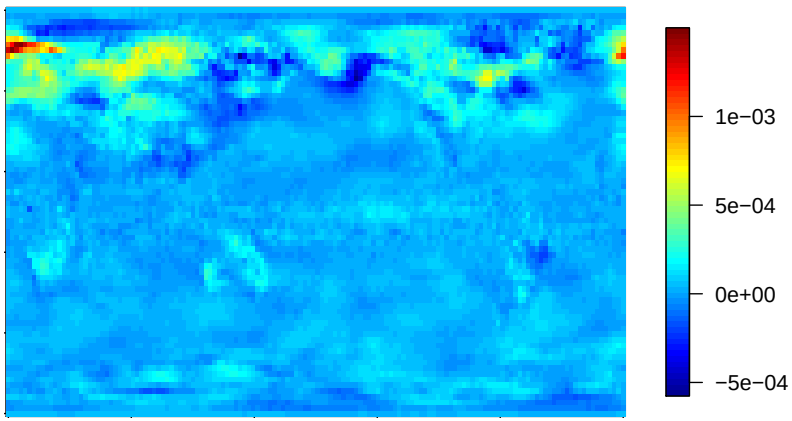
(b) Single machine: Column-wise compression ($MSE = 0.5661$)

Figure 8: Climate data: Comparison of different methods which return coefficients lying in the original space.

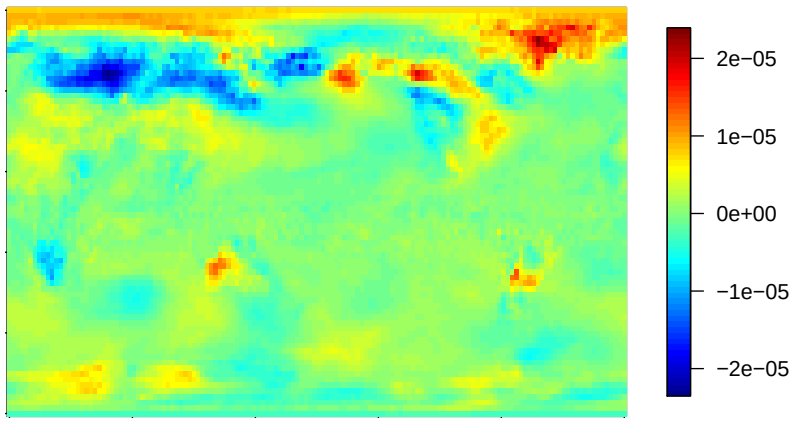
(a) The coefficients estimated in the non-distributed setting with SDCA
(b) The coefficients that are returned when the dimensionality of the design matrix is first compressed with a random projection prior to estimating the coefficients using SDCA in this low-dimensional space and then projected back to the original space (poor approximation, back-projection guaranteed to be a bad approximation (Zhang et al., 2012))

Experimental Results

Climate Data



(a) Single machine: Full solution ($MSE = 0.5756$)



(b) Single machine: Row-wise compression ($MSE = 0.9977$)

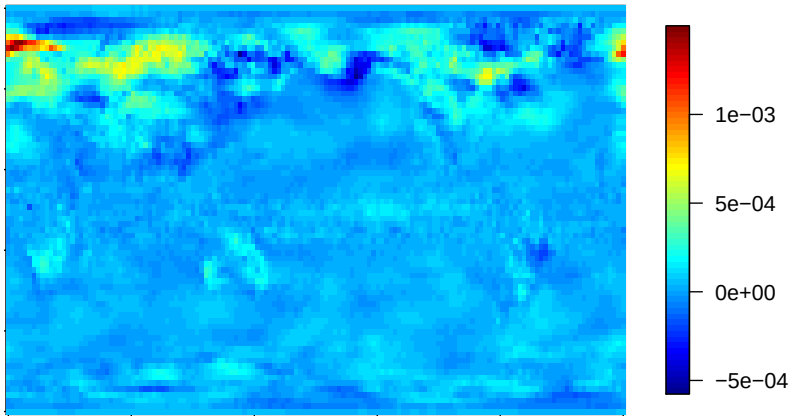
Figure 9: Climate data: Comparison of different methods which return coefficients lying in the original space.

(a) The coefficients estimated in the non-distributed setting with SDCA

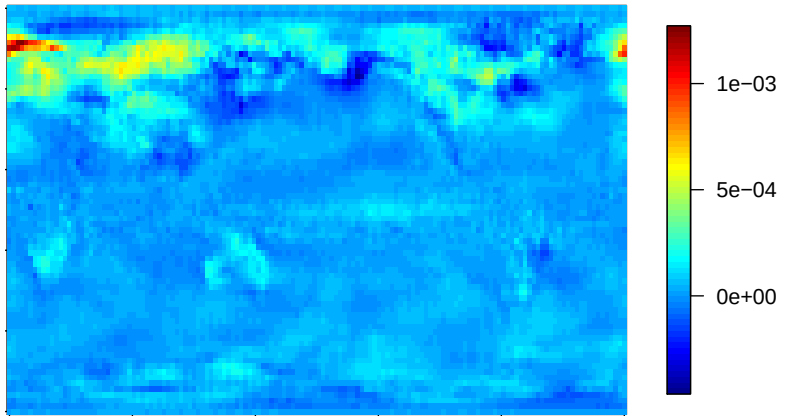
(b) The coefficients returned as a result of compressing the rows of $\mathbf{X}$ with a random projection to $n_{\text{subs}} = n/2$ prior to performing ridge regression (large MSE and poor approximation, large ratio between p and n, effective sample size reduced)

Experimental Results

Climate Data



(a) Single machine: Full solution ($MSE = 0.5756$)



(b) Distributed setting: Loco ($MSE = 0.5782$)

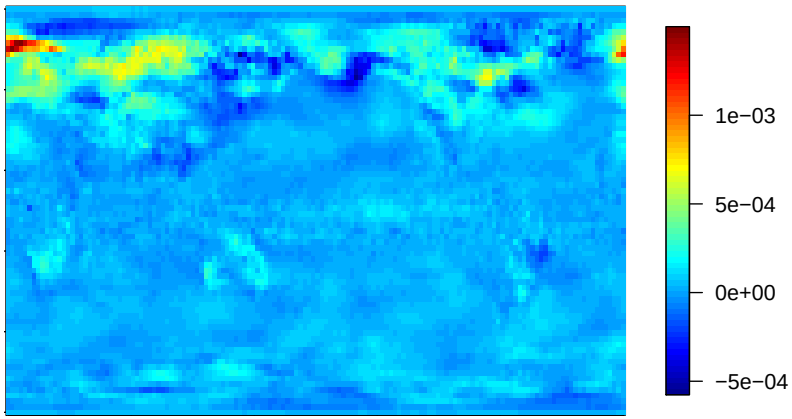
Figure 10: Climate data: Comparison of different methods which return coefficients lying in the original space.

(a) The coefficients estimated in the non-distributed setting with SDCA

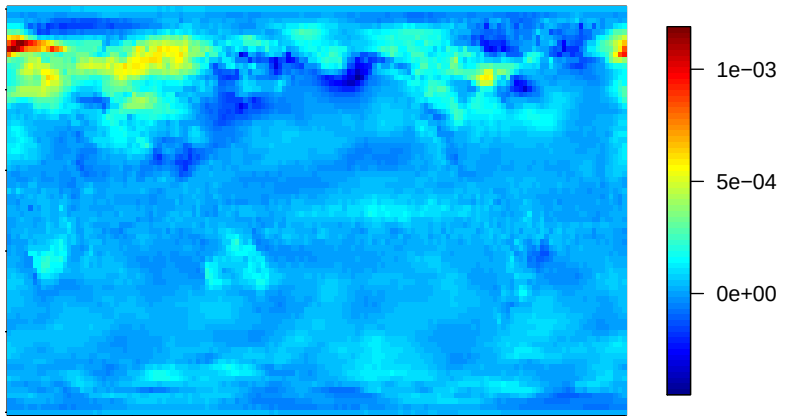
(b) The coefficients returned by Loco, distributed over 4 workers, compressing each worker's raw features ($\tau = 2592$) to 10% of the dimensionality, i.e. $\tau_{\text{subs}} = 260$ and concatenating these representations (similar to the optimal)

Experimental Results

Climate Data



(a) Single machine: Full solution ($MSE = 0.5756$)



(b) Distributed setting: CoCoA ($MSE = 0.5795$)

Figure 11: Climate data: Comparison of different methods which return coefficients lying in the original space.

(a) The coefficients estimated in the non-distributed setting with SDCA

(b) The coefficients returned by CoCoA (similar to the optimal)

Discussion

Summary

Loco:

- A simple algorithm for distributed ridge regression
- Requiring minimal communication and no synchronization
- Based on random projections
- Achieving small additional error compared with the optimal ridge regression solution
- Obtaining significant speedups with the number of workers
- Without making any additional assumptions about sparsity in the data
- Preserving structure in the estimated coefficients

Discussion

Future Work

Future Work:

- Use Loco with a sparse random projection if the data is very sparse, and see additional performance gains.
- Generalize Loco to a larger class of estimation problems (Dual-Loco (Heinze et al., 2015)).
- Explore the connection between Loco and privacy aware learning. Distributed optimization is a natural paradigm when preserving privacy is required. The class of J-L projections have been shown to preserve differential privacy (Blocki et al., 2012).
- Zhang et al. (2013a) have established bounds on the minimum amount of communication necessary for a distributed estimation task to achieve minimax optimal risk. It would be interesting to investigate how Loco fits into this framework since the distribution strategy of Loco differs from most commonly analysed methods.

References

- Achlioptas, Dimitris. "Database-friendly random projections: Johnson-Lindenstrauss with binary coins." *Journal of computer and System Sciences* 66.4 (2003): 671-687.
- Ailon, Nir, and Bernard Chazelle. "The fast Johnson-Lindenstrauss transform and approximate nearest neighbors." *SIAM Journal on Computing* 39.1 (2009): 302-322.
- Blocki, Jeremiah, et al. "The johnson-lindenstrauss transform itself preserves differential privacy." *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*. IEEE, 2012.
- Boutsidis, Christos, and Alex Gittens. "Improved matrix algorithms via the Subsampled Randomized Hadamard Transform. 2012." *arXiv preprint arXiv:1204.0062*.
- Duchi, John C., et al. "Information-theoretic lower bounds for distributed statistical estimation with communication constraints." *arXiv preprint arXiv:1405.0782* (2014).

References

- Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp. "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions." *SIAM review* 53.2 (2011): 217-288.
- Heinze, Christina, Brian McWilliams, and Nicolai Meinshausen. "DUAL-LOCO: Distributing Statistical Estimation Using Random Projections." *arXiv preprint arXiv:1506.02554* (2015).
- Heinze, Christina, et al. "LOCO: Distributing Ridge Regression with Random Projections." *arXiv preprint arXiv:1406.3469* (2014).
- Jaggi, Martin, et al. "Communication-efficient distributed dual coordinate ascent." *Advances in Neural Information Processing Systems*. 2014.
- Shalev-Shwartz, Shai, and Tong Zhang. "Stochastic dual coordinate ascent methods for regularized loss minimization." *Journal of Machine Learning Research* 14.Feb (2013): 567-599.
- Zhang, Lijun, et al. "Recovering the Optimal Solution by Dual Random Projection." *COLT*. 2013.