Information Processing Letters 114 (2014) 273–276

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A note on sparse least-squares regression

Christos Boutsidis [a,*], Malik Magdon-Ismail [b]

[a] Mathematical Sciences Department, IBM T.J. Watson Research Center, United States
[b] Computer Science Department, Rensselaer Polytechnic Institute, United States

ARTICLE INFO

Article history:
Received 29 May 2013
Received in revised form 2 September 2013
Accepted 15 November 2013
Available online 25 December 2013
Communicated by X. Wu

Keywords:
Algorithms
Least squares
Regression
Sparse approximation
Sparsification
Regularization
Truncated SVD

ABSTRACT

We compute a *sparse* solution to the classical least-squares problem $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2$, where $\mathbf{A}$ is an arbitrary matrix. We describe a novel algorithm for this sparse least-squares problem. The algorithm operates as follows: first, it selects columns from $\mathbf{A}$, and then solves a least-squares problem only with the selected columns. The column selection algorithm that we use is known to perform well for the well studied column subset selection problem. The contribution of this article is to show that it gives favorable results for sparse least-squares as well. Specifically, we prove that the solution vector obtained by our algorithm is close to the solution vector obtained via what is known as the "SVD-truncated regularization approach".

1. Introduction

Fix inputs $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. We study least-squares regression: $\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2$. It is well known that the minimum norm solution vector can be found using the pseudo-inverse of $\mathbf{A}$: $\mathbf{x}^* = \mathbf{A}^\dagger\mathbf{b} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. When $\mathbf{A}$ is ill-conditioned, $\mathbf{A}^\dagger$ becomes unstable to perturbations and overfitting can become a serious problem. For example, when the smallest non-zero singular value of $\mathbf{A}$ is close to zero, the largest singular value of $\mathbf{A}^\dagger$ can be extremely large and the solution vector $\mathbf{x}^* = \mathbf{A}^\dagger\mathbf{b}$ obtained via a numerical algorithm is not the optimal, due to numerical instability issues. Practitioners deal with such situations using *regularization*.

Popular regularization techniques are the Lasso [8], the Tikhonov regularization [4], and the truncated SVD [6]. The lasso minimizes $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 + \lambda\|\mathbf{x}\|_1$, and Tikhonov regularization minimizes $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$ (in both cases $\lambda > 0$ is the regularization parameter). The truncated SVD minimizes $\|\mathbf{A}_k\mathbf{x}-\mathbf{b}\|_2$, where $k < \text{rank}(\mathbf{A})$ is a rank parameter and $\mathbf{A}_k \in \mathbb{R}^{m\times n}$ is the best rank-k approximation to $\mathbf{A}$ obtained via the SVD. So, the truncated SVD solution is $\mathbf{x}_k^* = \mathbf{A}_k^\dagger\mathbf{b}$. Notice that these regularization methods impose parsimony on $\mathbf{x}$ in different ways. A combinatorial approach to regularization is to explicitly impose the sparsity constraint on $\mathbf{x}$, requiring it to have few non-zero elements. We give a new deterministic algorithm which, for $r = O(k)$, computes an $\hat{\mathbf{x}}_r \in \mathbb{R}^n$ with at most r non-zero entries such that $\|\mathbf{A}\hat{\mathbf{x}}_r - \mathbf{b}\|_2 \approx \|\mathbf{A}\mathbf{x}_k^* - \mathbf{b}\|_2$.

1.1. Preliminaries

The compact (or thin) Singular Value Decomposition (SVD) of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ of rank ρ is

$$\mathbf{A} = \underbrace{\begin{pmatrix}\mathbf{U}_k & \mathbf{U}_{\rho-k}\end{pmatrix}}_{\mathbf{U}_\mathbf{A}\in\mathbb{R}^{m\times\rho}} \underbrace{\begin{pmatrix}\boldsymbol{\Sigma}_k & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_{\rho-k}\end{pmatrix}}_{\boldsymbol{\Sigma}_\mathbf{A}\in\mathbb{R}^{\rho\times\rho}} \underbrace{\begin{pmatrix}\mathbf{V}_k^T\\ \mathbf{V}_{\rho-k}^T\end{pmatrix}}_{\mathbf{V}_\mathbf{A}^T\in\mathbb{R}^{\rho\times n}}.$$

Here, $\mathbf{U}_k \in \mathbb{R}^{m\times k}$ and $\mathbf{U}_{\rho-k} \in \mathbb{R}^{m\times(\rho-k)}$ contain the left singular vectors of $\mathbf{A}$. Similarly, $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ and $\mathbf{V}_{\rho-k} \in$

* Corresponding author.
E-mail addresses: cboutsi@us.ibm.com (C. Boutsidis), magdon@cs.rpi.edu (M. Magdon-Ismail).

0020-0190/$ – see front matter
http://dx.doi.org/10.1016/j.ipl.2013.11.011

Algorithm 1: Deterministic sparse regression

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, target rank $k < \text{rank}(\mathbf{A})$, and parameter $0 < \varepsilon < 1/2$.
Obtain $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ from the SVD of $\mathbf{A}$ and compute $\mathbf{E} = \mathbf{A} - \mathbf{A}\mathbf{V}_k\mathbf{V}_k^{\mathrm{T}} \in \mathbb{R}^{m\times n}$.
Set $\mathbf{C} = \mathbf{A}\boldsymbol{\Omega}\mathbf{S} \in \mathbb{R}^{m\times r}$, with $r = \lceil \frac{9k}{\varepsilon^2} \rceil$ and $[\boldsymbol{\Omega}, \mathbf{S}] = \text{DeterministicSampling}(\mathbf{V}_k^{\mathrm{T}}, \mathbf{E}, r)$,
Set $\mathbf{x}_r = \mathbf{C}^{\dagger}\mathbf{b} \in \mathbb{R}^r$, and $\hat{\mathbf{x}}_r = \boldsymbol{\Omega}\mathbf{S}\mathbf{x}_r \in \mathbb{R}^n$ ($\hat{\mathbf{x}}_r$ has at most r non-zeros at the indices of the selected columns in $\mathbf{C}$).
Return $\hat{\mathbf{x}}_r \in \mathbb{R}^n$.

$\mathbb{R}^{n\times(\rho-k)}$ contain the right singular vectors. The singular values of $\mathbf{A}$, which we denote as $\sigma_1(\mathbf{A}) \geqslant \sigma_2(\mathbf{A}) \geqslant \cdots \geqslant \sigma_\rho(\mathbf{A}) > 0$ are contained in $\boldsymbol{\Sigma}_k \in \mathbb{R}^{k\times k}$ and $\boldsymbol{\Sigma}_{\rho-k} \in \mathbb{R}^{(\rho-k)\times(\rho-k)}$. We use $\mathbf{A}^{\dagger} = \mathbf{V}_\mathbf{A}\boldsymbol{\Sigma}_\mathbf{A}^{-1}\mathbf{U}_\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ to denote the Moore–Penrose pseudo-inverse of $\mathbf{A}$ with $\boldsymbol{\Sigma}_\mathbf{A}^{-1}$ denoting the inverse of $\boldsymbol{\Sigma}_\mathbf{A}$. Let $\mathbf{A}_k = \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{V}_k^{\mathrm{T}} \in \mathbb{R}^{m\times n}$ and $\mathbf{A}_{\rho-k} = \mathbf{A} - \mathbf{A}_k = \mathbf{U}_{\rho-k}\boldsymbol{\Sigma}_{\rho-k}\mathbf{V}_{\rho-k}^{\mathrm{T}} \in \mathbb{R}^{m\times n}$. For $k < \text{rank}(\mathbf{A})$, the SVD gives the best rank k approximation to $\mathbf{A}$ in both the spectral and the Frobenius norm: for $\tilde{\mathbf{A}} \in \mathbb{R}^{m\times n}$, let $\text{rank}(\tilde{\mathbf{A}}) \leqslant k$; then, for $\xi = 2, \mathrm{F}$, $\|\mathbf{A} - \mathbf{A}_k\|_\xi \leqslant \|\mathbf{A} - \tilde{\mathbf{A}}\|_\xi$. Also, $\|\mathbf{A} - \mathbf{A}_k\|_2 = \|\boldsymbol{\Sigma}_{\rho-k}\|_2 = \sigma_{k+1}(\mathbf{A})$, and $\|\mathbf{A} - \mathbf{A}_k\|_{\mathrm{F}}^2 = \|\boldsymbol{\Sigma}_{\rho-k}\|_{\mathrm{F}}^2 = \sum_{i=k+1}^{\rho}\sigma_i^2(\mathbf{A})$. The Frobenius and the spectral norm of $\mathbf{A}$ are defined as: $\|\mathbf{A}\|_{\mathrm{F}}^2 = \sum_{i,j}\mathbf{A}_{ij}^2 = \sum_{i=1}^{\rho}\sigma_i^2(\mathbf{A})$; and $\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$. Let $\mathbf{X}$ and $\mathbf{Y}$ be matrices of appropriate dimensions; then, $\|\mathbf{X}\mathbf{Y}\|_{\mathrm{F}} \leqslant \min\{\|\mathbf{X}\|_{\mathrm{F}}\|\mathbf{Y}\|_2, \|\mathbf{X}\|_2\|\mathbf{Y}\|_{\mathrm{F}}\}$. This is a stronger version of the standard submultiplicativity property $\|\mathbf{X}\mathbf{Y}\|_{\mathrm{F}} \leqslant \|\mathbf{X}\|_{\mathrm{F}}\|\mathbf{Y}\|_{\mathrm{F}}$, which we will refer to as "spectral submultiplicativity".

Given $k < \rho = \text{rank}(\mathbf{A})$, the truncated rank-$k$ SVD regularized weights are

$$\mathbf{x}_k^* = \mathbf{A}_k^{\dagger}\mathbf{b} = \mathbf{V}_k\boldsymbol{\Sigma}_k^{-1}\mathbf{U}_k^{\mathrm{T}}\mathbf{b} \in \mathbb{R}^n,$$

and note that $\|\mathbf{b} - \mathbf{A}_k\mathbf{A}_k^{\dagger}\mathbf{b}\|_2 = \|\mathbf{b} - \mathbf{U}_k\mathbf{U}_k^{\mathrm{T}}\mathbf{b}\|_2$.

Finally, for $r < n$, let $\boldsymbol{\Omega} = [\mathbf{z}_{i_1}, \ldots, \mathbf{z}_{i_r}] \in \mathbb{R}^{n\times r}$ where $\mathbf{z}_i \in \mathbb{R}^m$ are standard basis vectors; $\boldsymbol{\Omega}$ is a *sampling matrix* because $\mathbf{A}\boldsymbol{\Omega} \in \mathbb{R}^{m\times r}$ is a matrix whose columns are sampled (with possible repetition) from the columns of $\mathbf{A}$. Let $\mathbf{S} \in \mathbb{R}^{r\times r}$ be a diagonal *rescaling matrix* with positive entries; then, we define the sampled and rescaled columns from $\mathbf{A}$ by $\mathbf{C} = \mathbf{A}\boldsymbol{\Omega}\mathbf{S}$: $\boldsymbol{\Omega}$ samples some columns from $\mathbf{A}$ and then $\mathbf{S}$ rescales them.

2. Results

Our sparse solver to minimize $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ takes as input the sparsity parameter r (i.e., the solution vector $\mathbf{x}$ is allowed at most r non-zero entries), and selects r rescaled columns from $\mathbf{A}$ (denoted by $\mathbf{C}$). We then solve the least-squares problem to minimize $\|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2$. The result is a dense vector $\mathbf{C}^{\dagger}\mathbf{b}$ with r dimensions. The sparse solution $\hat{\mathbf{x}}_r$ will be zero at indices corresponding to columns not selected in $\mathbf{C}$, and we use $\mathbf{C}^{\dagger}\mathbf{b}$ to compute the other entries of $\hat{\mathbf{x}}_r$.

Theorem 1. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, rank $k < \text{rank}(\mathbf{A})$, and $0 < \varepsilon < 1/2$. Algorithm 1 runs in time $O(mn\min\{m,n\} + nk^3/\varepsilon^2)$ and returns $\hat{\mathbf{x}}_r \in \mathbb{R}^n$ with at most $r = \lceil 9k/\varepsilon^2 \rceil$ non-zero entries such that:*

Algorithm 2: DeterministicSampling (from [1])

Input: $\mathbf{V}^{\mathrm{T}} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{k\times n}$; $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_n] \in \mathbb{R}^{m\times n}$; and $r > k$.
Output: Sampling and rescaling matrices $\boldsymbol{\Omega} \in \mathbb{R}^{n\times r}, \mathbf{S} \in \mathbb{R}^{r\times r}$.
Initialize $\mathbf{B}_0 = \mathbf{0}_{k\times k}$, $\boldsymbol{\Omega} = \mathbf{0}_{n\times r}$, $\mathbf{S} = \mathbf{0}_{r\times r}$.
for $\tau = 0$ **to** $r - 1$ **do**
  Set $\mathrm{L}_\tau = \tau - \sqrt{rk}$.
  Pick index $i \in \{1, 2, \ldots, n\}$ and t such that $U(\mathbf{e}_i) \leqslant \frac{1}{t} \leqslant L(\mathbf{v}_i, \mathbf{B}_\tau, \mathrm{L}_\tau)$.
  Update $\mathbf{B}_{\tau+1} = \mathbf{B}_\tau + t\mathbf{v}_i\mathbf{v}_i^{\mathrm{T}}$. Set $\boldsymbol{\Omega}_{i,\tau+1} = 1$ and $\mathbf{S}_{\tau+1,\tau+1} = 1/\sqrt{t}$.
end for
Return: $\boldsymbol{\Omega} \in \mathbb{R}^{n\times r}, \mathbf{S} \in \mathbb{R}^{r\times r}$.

$$\|\mathbf{A}\hat{\mathbf{x}}_r - \mathbf{b}\|_2 \leqslant \left\|\mathbf{A}\mathbf{x}_k^* - \mathbf{b}\right\|_2 + (1+\varepsilon)\cdot\|\mathbf{b}\|_2 \cdot \frac{\|\mathbf{A} - \mathbf{A}_k\|_{\mathrm{F}}}{\sigma_k(\mathbf{A})}.$$

This upper bound is "small" when $\mathbf{A}$ is "effectively" low-rank, i.e., $\|\mathbf{A} - \mathbf{A}_k\|_{\mathrm{F}}/\sigma_k(\mathbf{A}) \ll 1$. Also, a trivial bound is $\|\mathbf{A}\hat{\mathbf{x}}_r - \mathbf{b}\|_2 \leqslant \|\mathbf{b}\|_2$ (error when $\hat{\mathbf{x}}_r$ is the all-zeros vector), because $\|\mathbf{C}\mathbf{C}^{\dagger}\mathbf{b} - \mathbf{b}\|_2 \leqslant \|\mathbf{C}\mathbf{0}_{r\times 1} - \mathbf{b}\|_2 = \|\mathbf{b}\|_2$.

In the heart of Algorithm 1 lies a method for selecting columns from $\mathbf{A}$ (Algorithm 2), which was originally developed in [1] for column subset selection, where one selects columns $\mathbf{C}$ from $\mathbf{A}$ to minimize $\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\|_{\mathrm{F}}$. Here, we adopt the same algorithm for least-squares.

The main tool used to prove Theorem 1 is a new "structural" result that may be of independent interest.

Lemma 2. *Fix $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^n$, rank $k < \text{rank}(\mathbf{A})$, and sparsity $r > k$. Let $\mathbf{x}_k^* = \mathbf{A}_k^{\dagger}\mathbf{b} \in \mathbb{R}^n$, where $\mathbf{A}_k \in \mathbb{R}^{m\times n}$ is the rank-k SVD approximation to $\mathbf{A}$. Let $\boldsymbol{\Omega} \in \mathbb{R}^{n\times r}$ and $\mathbf{S} \in \mathbb{R}^{r\times r}$ be any sampling and rescaling matrices with $\text{rank}(\mathbf{V}_k^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{S}) = k$. Let $\mathbf{C} = \mathbf{A}\boldsymbol{\Omega}\mathbf{S} \in \mathbb{R}^{m\times r}$ be a matrix of sampled rescaled columns of $\mathbf{A}$ and let $\hat{\mathbf{x}}_r = \boldsymbol{\Omega}\mathbf{S}\mathbf{C}^{\dagger}\mathbf{b} \in \mathbb{R}^n$ (having at most r non-zeros). Then,*

$$\|\mathbf{A}\hat{\mathbf{x}}_r - \mathbf{b}\|_2 \leqslant \left\|\mathbf{A}\mathbf{x}_k^* - \mathbf{b}\right\|_2 + \left\|(\mathbf{A} - \mathbf{A}_k)\boldsymbol{\Omega}\mathbf{S}\left(\mathbf{V}_k^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{S}\right)^{\dagger}\boldsymbol{\Sigma}_k\mathbf{U}_k^{\mathrm{T}}\mathbf{b}\right\|_2.$$

The lemma says that if the sampling matrix satisfies a simple rank condition, then solving the regression on the sampled columns gives a sparse solution to the original problem with a performance guarantee.

2.1. Algorithm description

Algorithm 1 selects r columns from $\mathbf{A}$ to form $\mathbf{C}$ and the corresponding sparse vector $\hat{\mathbf{x}}_r$. The core of Algorithm 1 is the subroutine DeterministicSampling, which is a method to simultaneously sample the columns of two matrices, while controlling their spectral and Frobenius norms. DeterministicSampling takes inputs $\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{k\times n}$ and $\mathbf{E} \in \mathbb{R}^{m\times n}$; the matrix $\mathbf{V}$ is orthonormal, $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}_k$. (In our application, $\mathbf{V}^{\mathrm{T}} = \mathbf{V}_k^{\mathrm{T}}$ and $\mathbf{E} = \mathbf{A} - \mathbf{A}_k$.) We view $\mathbf{V}^{\mathrm{T}}$ and $\mathbf{E}$ as two sets of n column vectors, $\mathbf{V}^{\mathrm{T}} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$, and $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]$.

Given k and r and the iterator $\tau = 0, 1, 2, \ldots, r-1$, define $\mathrm{L}_\tau = \tau - \sqrt{rk}$. For a symmetric matrix $\mathbf{B} \in \mathbb{R}^{k\times k}$ with eigenvalues $\lambda_1, \ldots, \lambda_k$ and $\mathrm{L} \in \mathbb{R}$, define functions

$$\phi(\mathrm{L}, \mathbf{B}) = \sum_{i=1}^{k}\frac{1}{\lambda_i - \mathrm{L}},$$

and