Principal Component Projection Without Principal Component Analysis

Roy Frostig RF@CS.STANFORD.EDU
Stanford University

Cameron Musco CNMUSCO@MIT.EDU
Christopher Musco CPMUSCO@MIT.EDU
MIT

Aaron Sidford ASID@MICROSOFT.COM
Microsoft Research, New England

Abstract

We show how to efficiently project a vector onto the top principal components of a matrix, *without explicitly computing these components*. Specifically, we introduce an iterative algorithm that provably computes the projection using few calls to any black-box routine for ridge regression.

By avoiding explicit principal component analysis (PCA), our algorithm is the first with no runtime dependence on the number of top principal components. We show that it can be used to give a fast iterative method for the popular principal component regression problem, giving the first major runtime improvement over the naive method of combining PCA with regression.

To achieve our results, we first observe that ridge regression can be used to obtain a "smooth projection" onto the top principal components. We then sharpen this approximation to true projection using a low-degree polynomial approximation to the matrix step function. Step function approximation is a topic of long-term interest in scientific computing. We extend prior theory by constructing polynomials with simple iterative structure and rigorously analyzing their behavior under limited precision.

General background: Broad statement about data representation

1. Introduction

In machine learning and statistics, it is common—often essential—to represent data in a concise form that decreases noise and increases efficiency in downstream tasks.

Specific background: data projection via PCA is typical but expensive

Perhaps the most widespread method for doing so is to project data onto the linear subspace spanned by its directions of highest variance—that is, onto the span of the top components given by principal component analysis (PCA). Computing principal components can be an expensive task, a challenge that prompts a basic algorithmic question:

> *Can we project a vector onto the span of a matrix's top principal components without performing principal component analysis?*

Knowledge gap: Can we skip PCA?

This paper answers that question in the affirmative, demonstrating that projection is much easier than PCA itself. We show that it can be solved using a simple iterative algorithm based on black-box calls to a ridge regression routine. The algorithm's runtime *does not depend* on the number of top principal components chosen for projection, a cost inherent to any algorithm for PCA, or even algorithms that just compute an orthogonal span for the top components.

Here we show: Yes, can skip PCA by using an iterative ridge regression routine

1.1. Motivation: principal component regression

To motivate our projection problem, consider one of the most basic downstream applications for PCA: linear regression. Combined, PCA and regression comprise the *principal component regression* (PCR) problem:

Definition 1.1 (Principal component regression (PCR))**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be a design matrix whose rows are data points and let $\mathbf{b} \in \mathbb{R}^{n}$ be a vector of data labels. Let $\mathbf{A}_\lambda \in \mathbb{R}^{n\times d}$ denote the result of projecting each row of $\mathbf{A}$ onto the span of the top principal components of $\mathbf{A}$, i.e. the eigenvectors of the covariance matrix $\frac{1}{n}\mathbf{A}^\top\mathbf{A}$ whose corresponding eigenvalue exceeds a threshold λ. The task of PCR is to find a minimizer of the squared loss $\|\mathbf{A}_\lambda\mathbf{x} - \mathbf{b}\|_2^2$. In other words, the goal is to compute $\mathbf{A}_\lambda^\dagger\mathbf{b}$, where $\mathbf{A}_\lambda^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{A}_\lambda$.*

PCR is a key regularization method in statistics, numerical

NOTE: Paper expands on specific background to explain PCR

Proceedings of the 33^{rd} *International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

NOTE: Paper expands on specific background to explain ridge regression

linear algebra, and scientific disciplines including chemometrics (Hotelling, 1957; Hansen, 1987; Frank & Friedman, 1993). It models the assumption that small principal components represent noise rather than data signal. PCR is typically solved by first using PCA to compute $\mathbf{A}_\lambda$ and then applying linear regression. The PCA step dominates the algorithm's cost, especially if many principal components have variance above the threshold λ.

We remedy this issue by showing that our principal component *projection* algorithm yields a fast algorithm for *regression*. Specifically, full access to $\mathbf{A}_\lambda$ is unnecessary for PCR: $\mathbf{A}_\lambda^\dagger \mathbf{b}$ can be computed efficiently given only an approximate projection of the vector $\mathbf{A}^\mathsf{T}\mathbf{b}$ onto $\mathbf{A}$'s top principal components. By solving projection without PCA we obtain the first PCA-free algorithm for PCR.

1.2. A first approximation: ridge regression

Our approach to efficient principal component projection is actually based on a common alternative to PCR: ridge regression. This ubiquitous method computes a minimizer of $\|\mathbf{Ax}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$ for some regularization parameter λ (Tikhonov, 1963). The advantage of ridge regression is that it is a simple convex optimization problem that can be solved efficiently using many techniques (see Lemma 2.1).

Solving ridge regression is equivalent to applying the matrix $(\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\mathsf{T}$, an operation that can be viewed as a smooth relaxation of PCR. Adding the ℓ_2 norm penalty (i.e. $\lambda\mathbf{I}$) effectively "washes out" $\mathbf{A}$'s small principal components in comparison to its large ones and achieves an effect similar to PCR at the extreme ends of $\mathbf{A}$'s spectrum.

Accordingly, ridge regression gives access to a "smooth projection" operator, $(\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\mathsf{T}\mathbf{A}$, which approximates $\mathbf{P}_{\mathbf{A}_\lambda}$, the projection onto $\mathbf{A}$'s top row principal components. Both have the same singular vectors, but $\mathbf{P}_{\mathbf{A}_\lambda}$ has a singular value of 1 for each squared singular value $\sigma_i^2 \geq \lambda$ in $\mathbf{A}$ and a singular value of 0 for each $\sigma_i^2 < \lambda$, whereas $(\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\mathsf{T}\mathbf{A}$ has singular values equal to $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$. This function approaches 1 when σ_i^2 is much greater than λ and 0 when it is smaller. Figure 1 shows the comparison.

Unfortunately, in many settings, ridge regression is a very crude approximation to PCR and projection and may perform significantly worse in certain data analysis applications (Dhillon et al., 2013). In short, while ridge regression algorithms are valuable tools, it has been unclear how to wield them for tasks like projection or PCR.

1.3. Main result: from ridge regression to projection

We show that it is possible to *sharpen* the weak approximation given by ridge regression. Specifically, there exists a low degree polynomial $p(\cdot)$ such that $p\left((\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\mathsf{T}\mathbf{A}\right)\mathbf{y}$ provides a very accurate approximation to $\mathbf{P}_{\mathbf{A}_\lambda}\mathbf{y}$ for any $\mathbf{y}\in\mathbb{R}^d$. Moreover, the polynomial can be evaluated as a recurrence, which translates into a simple iterative algorithm: we can apply the sharpened approximation to a vector by repeatedly applying any ridge regression routine a small number of times.

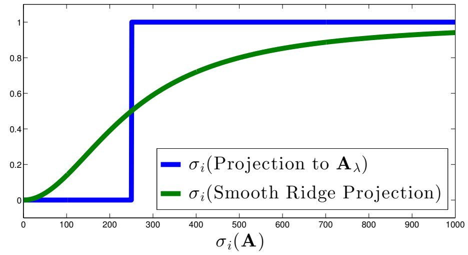

Figure 1: Singular values of the projection matrix $\mathbf{P}_{\mathbf{A}_\lambda}$ vs. those of the smooth projection operator $(\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}\mathbf{A}^\mathsf{T}\mathbf{A}$ obtained from ridge regression.

Theorem 1.2 (Principal component projection without PCA)**.** *Given* $\mathbf{A}\in\mathbb{R}^{n\times d}$ *and* $\mathbf{y}\in\mathbb{R}^d$*, Algorithm 1 uses* $\tilde{O}(\gamma^{-2}\log(1/\epsilon))$ *approximate applications of* $(\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}$ *and returns* $\mathbf{x}$ *with* $\|\mathbf{x}-\mathbf{P}_{\mathbf{A}_\lambda}\mathbf{y}\|_2 \leq \epsilon\|\mathbf{y}\|_2$.

Like all iterative PCA algorithms, our running time scales inversely with γ, the *spectral gap* around λ.[1] Notably, it does not depend on the number of principal components in $\mathbf{A}_\lambda$, a cost incurred by any method that applies the projection $\mathbf{P}_{\mathbf{A}_\lambda}$ directly, either by explicitly computing the top principal components of $\mathbf{A}$, or even by just computing an orthogonal span for these components.

As mentioned, the above theorem also yields an algorithm for principal component *regression* that computes $\mathbf{A}_\lambda^\dagger\mathbf{b}$ without finding $\mathbf{A}_\lambda$. We achieve this result by introducing a robust reduction, from projection to PCR, that again relies on ridge regression as a computational primitive.

Corollary 1.3 (Principal component regression without PCA)**.** *Given* $\mathbf{A}\in\mathbb{R}^{n\times d}$ *and* $\mathbf{b}\in\mathbb{R}^n$*, Algorithm 2 uses* $\tilde{O}(\gamma^{-2}\log(1/\epsilon))$ *approximate applications of* $(\mathbf{A}^\mathsf{T}\mathbf{A}+\lambda\mathbf{I})^{-1}$ *and returns* $\mathbf{x}$ *with* $\|\mathbf{x}-\mathbf{A}_\lambda^\dagger\mathbf{b}\|_{\mathbf{A}^\mathsf{T}\mathbf{A}} \leq \epsilon\|\mathbf{b}\|_2$.

Corollary 1.3 gives the first known algorithm for PCR that avoids the cost of principal component analysis.

1.4. Related work

A number of papers attempt to alleviate the high cost of principal component analysis when solving PCR. It has

[1] See Section 3.2 for a discussion of this gap dependence. Aside from a full SVD requiring $O(nd^2)$ time, any PCA algorithm giving the guarantee of Theorem 1.2 will have a dependence both on γ *and* on the number principal components in $\mathbf{A}_\lambda$. However, the γ dependence can be better – $\gamma^{-1/2}$ for Krylov methods (Musco & Musco, 2015), giving a runtime tradeoff.

NOTE: Further explanation of knowledge gap, i.e., how to use ridge regression.

NOTE: Paper expands on main result to detail theorem and corollary

NOTE: In this paper, related work section separated from rest of intro. In some papers, the related works are integrated into general intro structure.

been shown that an approximation to $\mathbf{A}_\lambda$ suffices for solving the regression problem (Chan & Hansen, 1990; Boutsidis & Magdon-Ismail, 2014). Unfortunately, even the fastest approximations are much slower than routines for ridge regression and inherently incur a linear dependence on the number of principal components above λ.

More closely related to our approach is work on the *matrix sign function*, an important operation in control theory, quantum chromodynamics, and scientific computing in general. Approximating the sign function often involves matrix polynomials similar to our "sharpening polynomial" that converts ridge regression to principal component projection. Significant effort addresses Krylov methods for applying such operators without computing them explicitly (van den Eshof et al., 2002; Frommer & Simoncini, 2008).

Our work differs from these methods in an important way: since we only assume access to an approximate ridge regression algorithm, it is essential that our sharpening step is robust to noise. Our iterative polynomial construction allows for a complete and rigorous noise analysis that is not available for Krylov methods, while at the same time eliminating space and post-processing costs. Iterative approximations to the matrix sign function have been proposed, but lack rigorous noise analysis (Higham, 2008).