An Efficient Algorithm for Low-Rank Kernel Regression
CS 229T Project Report
Hamidreza Hakim Javadi and Yash Deshpande
March 14, 2014

Abstract

Kernel regression is widely-used nonparametric technique in statistics for constructing predictors in a purely data-driven fashion. Due to the typical size of modern day datasets, it is natural to attempt to solve the kernel regression optimization through the application of a first-order method, with a hope to reduce complexity below the quadratic time barrier. However, even the evaluation of the full kernel matrix (and its application) places a bottleneck to this program of reducing computational complexity. We propose an algorithm that employs sparse evaluations of the kernel matrix, thereby greatly reducing the kernel evaluation as well as iteration cost. In the case of perfectly low-rank kernels, we rigorously prove that this algorithm allows to achieve essentially equivalent predictive performance, with far fewer kernel evaluations.

1 Introduction

Kernel methods are a nonparametric regression technique that allow the addition of richer, “non-linear” explanatory features while still retaining the computational and theoretical advantages of linear regression techniques. When responses are parameters on the real line, the kernel regression problem reduces to solving an optimization of the following sort.

$$\min_{u \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, (Ku)_i) + \lambda \langle u, Ku \rangle.$$  

Here \(\{x_i, y_i\}_{1 \leq i \leq n}\) are the training examples, \(K\) is the kernel matrix with entries \(K_{ij} = k(x_i, x_j)\) and \(\ell(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}\) is the loss function, typically chosen to be convex in its second argument. In this report, we will focus on the case of kernel ridge regression, namely the case of the squared loss. Letting \(y \in \mathbb{R}^n\) denote the vector of responses, we consider the following optimization problem:

$$\min_{u \in \mathbb{R}^n} f(u) \equiv \|y - Ku\|_2^2 + \lambda \langle u, Ku \rangle.$$  

This problem has an explicit least squares solution: \(Ku^* = (K + \lambda I_n)^{-1}Ky\). Here, in the case of rank-deficient \(K\), it suffices (for optimization training error) to look only at predictors in range space of \(K\). For explicit optimization, hence we require to evaluate the full kernel matrix \(K\) and solve a linear system with the regularized version \((K + \lambda I_n)\). When kernel evaluations are expensive it is natural to approximate the kernel matrix with a version that involves far fewer than \(n^2\) evaluations, but still retains the prediction accuracy required. Indeed, [Bac12] proposes just such a method based on randomized column sampling and proves, for kernels of rank \(r\) that requires \(O(nr)\) kernel evaluations but employs solving a linear system of equations with the approximation, hence \(O(r^2n + r^3)\) complexity. We relax slightly the problem: for an optimization gap of size \(\varepsilon > 0\) we propose an algorithm that provably requires \(O(nr/\varepsilon^2)\) operations and \(O(nr/\varepsilon^2)\) kernel evaluations. Our simulations, however, show a performance that is significantly better than these bounds for well-conditioned problems.
2 Algorithm and main result

The standard gradient descent algorithm for solving the optimization problem 1 as follows:

\[ u_{t+1} = u_t - \gamma_t K ((K + \lambda I_n)u_t - y) \]

\[ \hat{u}_t = \left( \sum_{t \leq \ell} \gamma_{\ell} \right)^{-1} \sum_{t \leq \ell} \gamma_{\ell} u_{\ell}. \]

Here \( \gamma_t \) is an adjustable learning rate. In this version, we replace \( K \) with independent sparsified versions \( K_1^t, K_2^t \) as follows:

\[ u_{t+1} = u_t - \gamma K_1^t ((K_2^t + \lambda I_n)u_t - y). \]  \hspace{1cm} (2)

The running average \( \hat{u}_t \) is defined previously. The key idea here is that when the kernel matrix \( K \) is low rank (or even well approximated by a low rank matrix), a sparsified version of \( K \) retains most of this linear structure. Hence the algorithm modification above essentially behaves as a noisy or stochastic gradient descent. Indeed, our proof of the algorithm performance relies on this observation. Our main result is as follows:

**Theorem 1.** Consider a kernel matrix \( K \) which has rank \( r \) and satisfies \( \| K \|_2 \leq M \). Assume that the observations \( y \) and at least one solution \( u^* \) satisfy \( \max(\| y \|_2, \| u^* \|) \leq R \). Then, there exist universal constants \( C, C_1 \) such that the following happens for \( n \) large enough. For any \( p \geq C r/n \), with \( n^2 p \) kernel evaluations per iteration and \( C_1 M^2 R^3 (1 + \lambda)^2 / \epsilon^2 \sqrt{mp} \) iterations of (2) with \( \gamma_t = C_1^{-1} \epsilon \sqrt{mp}/M^2 R^2 (1 + \lambda)^2 \) we have that \( \mathbb{E}\{f(u_T)\} - \text{Opt} \leq \epsilon \).

A few remarks are in order.

**Remark 2.1.** The presence of a bound on the size of \( y \) and \( u^* \) is not surprising. Our algorithm is a first-order method and the standard analyses [BTN01] of such methods scale in proportion with the size of the search space (here a ball of radius \( R \)).

**Remark 2.2.** The error bound in the objective decays as \( 1/\sqrt{t} \) where \( t \) is the number of iterations. With further assumptions on the condition number of \( K \), we believe this can be improved to \( 1/t \) since orthogonal to the kernel of \( K \), the objective function is strongly convex. Our simulations support this claim.

The key portion of our algorithm analysis involves analysing the spectral norm of the random matrix \( \| K_1^t - K \|_2 \). Here we borrow on tools developed from the celebrated result of Friedman, Kahn and Szemeredi in [FKS89], and further generalized in [FO05], [KMO10]. Indeed, we use a principal result of [KMO10] to prove a key bound on the second moment of \( \| K_1^t - K \|_2 \), which allows to bound the variance in the gradient.

3 Simulations and Examples

In this section we run the algorithm above on synthetic data described below and empirically demonstrate its convergence properties. In this simulation, we generate \( n = 500 \) training examples \( x_i \)'s in \( \mathbb{R}^d \), \( d = 10 \). \( x_i \)'s are iid \( \mathcal{N}(0, I_d) \) random vectors. Also, \( y_i \)'s are iid \( \mathcal{N}(0, 1) \) random variables. In this simulation, we use the linear kernel, we set \( p = 0.05 \), therefore, the expected number of nonzero elements in each row is 25. In addition, in order to remove the effect of high degree nodes we perform trimming. In other words, we set the rows that have more than 25 nonzero elements to zero. In the following figures we can see the excess cost \( = f(u_t) - f(u^*) \), versus number of iterations in logarithmic scale. Figure 1 shows the results for a single realization of data. Figure 2 is the average of 50 different realizations. It can be seen that the slope of the curve is about 1, which is a better result than the Theorem 1.

It is possible to generalize these results to other kernels like gaussian kernel in some cases. As an example, if \( x_i \)'s and \( y_i \)'s are generated as above, it can be seen that the diagonal entries of the gaussian kernel matrix
are equal to 1 and the other entries are almost equal to $b$, for some $b < 1$. This matrix has a large eigenvalue $\lambda_1 = (n-1)b + 1$ and $n-1$ small eigenvalues $\lambda_i = 1 - b, i = 2, \ldots, n$. Therefore, the kernel matrix in this case is approximately a low rank matrix and we can expect that the algorithm will work in this situation.

4 Proof of Theorem 1

Our proof of theorem 1 involves the following key results of [KMO10], [BTN01]. The results below are restated to correspond to our setting, and follow immediately from the original theorems.

Proposition 4.1 (Keshavan, Montanari, Oh). For a rank $r$ matrix $X$ with norm at most $M$, denote by $X_E$ the matrix wherein entries outside a subset $E \in [n] \times [n]$ are set to zero. Further, let $\mathcal{P}_r(M)$ denote the rank-$r$ projection of $M$. Let $E$ be chosen in the following manner. First keep each entry (up to symmetry) $i,j \in E$ with a probability $p \geq Cr/n$. Thereafter, remove from $E$ all rows (and columns) that have more
than 2np entries. Then, with probability at least \( 1 - 1/n^3 \)
\[
\left\| X - \frac{1}{p} \mathcal{P}_p(X^E) \right\|_2 \leq \frac{C'M}{\sqrt{np}}.
\]
for some universal constant \( C' \).

**Proposition 4.2** (Nemirovski). Under the assumptions of Theorem 1, suppose further that the gradient estimates \( g(u) \) satisfy \( \mathbb{E}\{\|g(u) - f'(u)\|_2^2\} \leq L^2 \). Then with \( \gamma_t = \sqrt{2R/L} \sqrt{N} \) we have that:
\[
\mathbb{E}\{f(u_N)\} - \text{Opt} \leq \frac{\sqrt{2RL}}{\sqrt{N}}.
\]

We use the following lemma to bound the gradient error due to sparsification. The result then follows by a direct application of Proposition 4.2.

**Lemma 4.3.** Consider the iteration Eq. (2), which employs the gradient approximation
\[
g(u) = K_1^T(\lambda u - y) + K_2^T u.
\]
Here \( K_1^T, K_2^T \) are independent sparsifications of the matrix \( K \), i.e.
\[
(K^T_1)_{ij} = \begin{cases} 
K_{ij}/p & \text{with probability } p. \\
0 & \text{otherwise.}
\end{cases}
\]
Then \( \mathbb{E}\{g(u)\} = f'(u) = K(u - y) + K^2u \) and \( \mathbb{E}\{\|g(u) - f'(u)\|_2^2\} \leq (C_1(1 + \lambda)MR/(np)^{1/4})^2 \).

## 5 Future work

We list below some open directions for future work, which we believe are interesting:

**Optimal convergence** The \( 1/\sqrt{T} \) convergence we use here is not displayed in the simulations, particularly for well conditioned problems. It appears that here a \( 1/T \) rate can be achieved under condition number guarantees.

**Connection with random graphs** We reuse, in our analysis, the bound for \( K_1^T - K \) in understanding the error \( K_1^T K_2^T - K^2 \). A heuristic calculation shows that this yields a loss of \( 1/\sqrt{np} \) in the error, which can possibly improve Theorem 1 above. This exploits a connection of the product matrix with a certain “product” random graph.

## References


A Proof of Lemma 4.3

Proof. The fact that $E\{g(u)\} = f'(u)$ follows directly from the sparsification scheme and the independence of $K_1^t, K_2^t$. As for the variance, we have:

$$E\{\|g(u) - f'(u)\|_2^2\} = E\{\|(K_1^t - K)(\lambda u - y) + (K_1^t K_2^t - K^2) y\|\}$$

$$\leq 2E \left\{ \|K_1^t - K\|_2^2 \right\} (\lambda + 1)^2 R^2 + 2E \left\{ \|K_1^t K_2^t - K^2\|_2^2 \right\} R^2,$$

where the inequality follows from the definition of the spectral norm and triangle inequality. We first bound $E\{\|K_1^t - K\|_2^2\}$ as follows. By Proposition 4.1 with high probability we have that $\|K_1^t - K\|_2^2 \leq C'M/\sqrt{np}$. Further, on the complement event (of probability at most 1/n^3) we have that $\|K_1^t - K\|_2^2 \leq (Mr/p)^2$. Consequently:

$$E\{\|K_1^t - K\|_2^2\} \leq \frac{C^2 M^2}{np} + \frac{M^2 r^2}{n^3 p^2}.$$

Since $p \geq Cr/n$ the second term is of lower order. Consequently for $n$ large enough:

$$E\{\|K_1^t - K\|_2^2\} \leq \left( \frac{C'M}{\sqrt{np}} \right)^2,$$

for some adjusted universal constant $C'$.

As for the second term, we write:

$$K_1^t K_2^t - K_2 = (K_1^t - K)(K_2^t - K) + (K_1^t - K)K + K(K_2^t - K).$$

Hence by triangle inequality:

$$\|K_1^t K_2^t - K\|_2 \leq \|K_1^t - K\|_2 \|K_2^t - K\|_2 + M(\|K_1^t - K\|_2 + \|K_2^t - K\|_2).$$

By a similar argument as before, and using $(a + b)^2 \leq 2(a^2 + b^2)$:

$$E\{\|K_1^t K_2^t - K\|_2^2\} \leq 2 \left( \frac{C'M}{\sqrt{np}} \right)^2 + 4 \frac{C'M^2}{\sqrt{np}}.$$

Since $np \geq Cr$ for a numerical constant $C$ and $r \geq 1$, we have that:

$$E\{\|K_1^t K_2^t - K\|_2^2\} \leq \frac{C'' M^2}{\sqrt{np}}.$$

Hence, we have that:

$$E\{\|g(u) - f'(u)\|_2^2\} \leq \frac{C'' M^2 R^2}{\sqrt{np}} ((\lambda + 1)^2 + \frac{1}{\sqrt{np}})$$

$$\leq \frac{C_1^2 M^2 R^2 (1 + \lambda)^2}{\sqrt{np}}.$$

Here $C_1$ is a large enough universal constant. \qed