NysADMM: faster composite convex optimization via low-rank approximation

Shipu Zhao [*] [1] **Zachary Frangella** [*] [2] **Madeleine Udell** [2]

Abstract

This paper develops a scalable new algorithm, called NysADMM, to minimize a smooth convex loss function with a convex regularizer. NysADMM accelerates the inexact Alternating Direction Method of Multipliers (ADMM) by constructing a preconditioner for the ADMM subproblem from a randomized low-rank Nyström approximation. NysADMM comes with strong theoretical guarantees: it solves the ADMM subproblem in a constant number of iterations when the rank of the Nyström approximation is the effective dimension of the subproblem regularized Gram matrix. In practice, ranks much smaller than the effective dimension can succeed, so NysADMM uses an adaptive strategy to choose the rank that enjoys analogous guarantees. Numerical experiments on real-world datasets demonstrate that NysADMM can solve important applications, such as the lasso, logistic regression, and support vector machines, in half the time (or less) required by standard solvers. The breadth of problems on which NysADMM beats standard solvers is a surprise: it suggests that ADMM is a dominant paradigm for numerical optimization across a wide range of statistical learning problems that are usually solved with bespoke methods.

1. Introduction

Consider the composite convex optimization problem

$$\text{minimize}_{x\in\mathbb{R}^d}\ \ell(Ax;b)+r(x). \tag{1}$$

We assume that ℓ and r are convex and ℓ is smooth. In machine learning, generally ℓ is a loss function, r is a regularizer, $A\in\mathbb{R}^{n\times d}$ is a feature matrix, and $b\in\mathbb{R}^n$ is the label or response. Throughout the paper we assume that a solution to (1) exists. A canonical example of (1) is the lasso problem,

$$\text{minimize}\ \frac{1}{2}\|Ax-b\|_2^2+\gamma\|x\|_1, \tag{2}$$

where $\ell(Ax;b)=\frac{1}{2}\|Ax-b\|_2^2$ and $r(x)=\gamma\|x\|_1$. We discuss more applications of (1) in Section 3.

The alternating directions method of multipliers (ADMM) is a popular algorithm to solve optimization problems of the form (1). However, when the matrix A is large, each iteration of ADMM requires solving a large subproblem. For example, consider the lasso where the loss ℓ is quadratic. At each iteration, ADMM solves a regularized least-squares problem at a cost of $O(nd^2)$ flops. On the other hand, it is not necessary to solve each subproblem exactly to ensure convergence: ADMM strategies that solve the subproblems inexactly are called inexact ADMM, and can be shown to converge when the sequence of errors is summable (Eckstein & Bertsekas, 1992). Unfortunately, it can be challenging even to satisfy this relaxed criterion. Consider again the lasso problem. At each iteration, inexact ADMM solves the regularized least-squares subproblem (4) approximately, for example, using the iterative method of conjugate gradients (CG). We call this method inexact ADMM with CG. The number of CG iterations required to achieve accuracy ϵ increases with the square root of the condition number κ_2 of the regularized Hessian, $O\left(\sqrt{\kappa_2}\log(\frac{\kappa_2}{\epsilon})\right)$. Alas, the condition number of large-scale data matrices is generally high, and later iterations of inexact ADMM require high accuracy, so inexact ADMM with CG still converges too slowly to be practical.

In this work we show how to speed up inexact ADMM using preconditioned conjugate gradients (PCG) as a subproblem solver. We precondition with randomized Nyström preconditioning (Frangella et al., 2021), a technique inspired by recent developments in randomized numerical linear algebra (RandNLA). We call the resulting algorithm NysADMM ("nice ADMM"): inexact ADMM with PCG using randomized Nyström preconditioning. The Nyström preconditioner reduces the number of iterations required to solve the subproblem to ϵ-accuracy to $O\left(\log(\frac{1}{\epsilon})\right)$, independent of the condition number. For non-quadratic loss functions, NysADMM uses linearized inexact ADMM and accelerates the linear subproblem solve similarly.

[*]Equal contribution [1]Cornell University, Ithaca, NY, USA. [2]Stanford University, Stanford, CA, USA. Correspondence to: Shipu Zhao <sz533@cornell.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

1.1. Contributions

1. We provide a general algorithmic framework for solving large scale lasso, ℓ_1-regularized logistic regression, and SVM problems.

2. Our theory shows that at each iteration only a constant number of matrix vector products (matvecs) are required to solve the ADMM subproblem, provided we have constructed the preconditioner appropriately. If the loss function is quadratic, only a constant number of matvecs are required to achieve convergence.

3. We develop a practical adaptive algorithm that increases the rank until the conditions of our theory are met, which ensures the theoretical benefits of the method can be realized in practice.

4. Even a preconditioner with lower rank often succeeds in speeding up inexact ADMM with PCG. Our analysis is also able to explain this phenomenon.

5. Our algorithm beats standard solvers such as glmnet, SAGA, and LIBSVM on large dense problems like lasso, logistic regression, and kernalized SVMs: it yields equally accurate solutions and often runs 2–4 times faster.

1.2. Related work

Our work relies on recent advancements in RandNLA for solving regularized least squares problems $(A^TA+\mu I)x = A^Tb$ for x, given a design matrix $A \in \mathbb{R}^{n\times d}$, righthand side $b \in \mathbb{R}^n$, and regularization $\mu \in \mathbb{R}$, using a *sketch* of the design matrix A (Lacotte & Pilanci, 2020). NysADMM adapts the randomized Nyström preconditioner of (Frangella et al., 2021). These algorithms begin by forming a *sketch* $Y = A\Omega$ of A (or A^T) with a random dimension reduction map $\Omega \in \mathbb{R}^{d\times s}$ (Martinsson & Tropp, 2020; Woodruff, 2014). For example, Ω may be chosen to have iid Gaussian entries. These algorithms obtain significant computational speedups by using a sketch size $s \ll \min\{n, d\}$ and working with the sketch in place of the original matrix to construct a preconditioner for the linear system. Frangella et al. (2021) and Lacotte & Pilanci (2020) show that these randomized preconditioners work well when the sketch size grows with the *effective dimension* (Equation (8)) of the Gram matrix (assuming, for Lacotte & Pilanci (2020) that we have access to a matrix square root). As the effective dimension is never larger than d and often significantly smaller, these results substantially improve on prior work in randomized preconditioning (Meng et al., 2014; Rokhlin & Tygert, 2008) that requires a sketch size $s \gtrsim d$. Many applications require even smaller sketch sizes: for example, for NysADMM, a fixed sketch size $s = 50$ suffices even for extremely large problems.

We are not the first to use RandNLA to accelerate iterative optimization. Gower et al. (2019); Pilanci & Wainwright (2017) both use iterative sketching to accelerate Newton's method, while Chowdhuri et al. (2020) use randomized preconditioning to accelerate interior point methods for linear programmming. The approach taken here is closest in spirit to (Chowdhuri et al., 2020), as we also use randomized preconditioning. However, the preconditioner used in (Chowdhuri et al., 2020) requires the data matrix to have many more columns than rows, while ours can handle any (sufficiently large) dimensions.

NysADMM can solve many traditional machine learning problems, such as lasso, regularized logistic regression, and support vector machines (SVMs). In contrast, standard solvers for these problems use a wider variety of convex optimization techniques. For example, one popular lasso solver, glmnet (Friedman et al., 2010), relies on coordinate descent (CD), while solvers for SVMs, such as LIBSVM (Chang & Lin, 2011), more often use sequential minimal optimization (Platt, 1998), a kind of pairwise CD on the dual problem. For regularized logistic regression, especially for ℓ_1 regularization, stochastic gradient algorithms are most commonly used (Defazio et al., 2014; Schmidt et al., 2017). Other authors propose to solve lasso with ADMM (Boyd et al., 2011; Yue et al., 2018). Our work, motivated by the ADMM quadratic programming framework of Stellato et al. (2020), is the first to accelerate ADMM with randomized preconditioning, thereby improving on the performance of standard CD or stochastic gradient solvers for each of these important classes of machine learning problems on large-scale dense data. Unlike Stellato et al. (2020), our work relies on inexact ADMM and can handle non-quadratic loss functions, which allows NysADMM to solve problems such as regularized logistic regression.

1.3. Organization of the paper

Section 2 introduces the NysADMM algorithm and necessary background from RandNLA. Section 3 lists a variety of applied problems that can be solved by NysADMM. Section 4 states the theoretical guarantees for NysADMM. Section 5 compares NysADMM and standard optimization solvers numerically on several applied problems. Section 6 summarizes the results of the paper and discusses directions for future work.

1.4. Notation and preliminaries

We call a matrix psd if it is positive semidefinite. The notation $a \gtrsim b$ means that $a \geq Cb$ for some absolute constant C. Given a matrix H, we denote its spectral norm by $\|H\|$. We denote the Moore-Penrose pseudoinverse of a matrix M by $M^\dagger$. For $\rho > 0$ and a symmetric psd matrix H, we define $H_\rho = H + \rho I$. We say a positive sequence

$\{\varepsilon^k\}_{k=1}^{\infty}$ is summable if $\sum_{k=1}^{\infty} \varepsilon^k < \infty$. We denote the Loewner ordering on the cone of symmetric psd matrices by $\preceq$, that is $A \preceq B$ if and only if $B - A$ is psd.

2. Algorithm

2.1. Inexact linearized ADMM

To solve problem (1), we apply the ADMM framework. Algorithm 1 shows the standard ADMM updates, where the regularizer $r = g + h$ is split into a smooth part g and a nonsmooth part h.

```
Algorithm 1 ADMM
input  feature matrix $A$, response $b$, loss function $\ell$, regularization $g$ and $h$, stepsize $\rho$
  repeat
    $x^{k+1} = \text{argmin}_x\{\ell(Ax;b) + g(x) + \frac{\rho}{2}\|x - z^k + u^k\|_2^2\}$
    $z^{k+1} = \text{argmin}_z\{h(z) + \frac{\rho}{2}\|x^{k+1} - z + u^k\|_2^2\}$
    $u^{k+1} = u^k + x^{k+1} - z^{k+1}$
  until convergence
output  solution $x_\star$ of problem (1)
```

In each iteration, two subproblems are solved sequentially to update variables x and z. The z-subproblem often has a closed-form solution. For example, if $h(x) = \|x\|_1$, the z-subproblem is the soft thresholding, and if h is the indicator function of a convex set $\mathcal{C}$, the z-subproblem is projection onto the set $\mathcal{C}$.

There is usually no closed-form solution for the x-subproblem. Instead, it is usually solved inaccurately by an iterative scheme, especially for large-scale applications. To simplify the subproblem, inspired by linearized ADMM, we assume ℓ and g are twice differentiable and notice that the x update is close to the minimum of a quadratic function given by the Taylor expansion of ℓ and g at the current iterate:

$$\begin{aligned}
&\tilde{x}^{k+1} = \text{argmin}_x\{\ell(A\tilde{x}^k;b) + (x - \tilde{x}^k)^T A^T \nabla \ell(A\tilde{x}^k;b) \\
&+ \frac{1}{2}(x - \tilde{x}^k)^T A^T H^\ell(A\tilde{x}^k;b)A(x - \tilde{x}^k) + g(\tilde{x}^k) \\
&+ (x - \tilde{x}^k)^T \nabla g(\tilde{x}^k) + \frac{1}{2}(x - \tilde{x}^k)^T H^g(\tilde{x}^k)(x - \tilde{x}^k) \\
&+ \frac{\rho}{2}\|x - \tilde{z}^k + \tilde{u}^k\|_2^2\}.
\end{aligned} \tag{3}$$

Here H^ℓ and H^g are the Hessian of ℓ and g respectively. We assume throughout the paper that H^ℓ and H^g are psd matrices, this is a very minor assumption, and is satisfied by all the applications we consider. The solution to this quadratic minimization may be obtained by solving the linear system

$$(A^T H^\ell(A\tilde{x}^k;b)A + H^g(\tilde{x}^k) + \rho I)x = r^k \tag{4}$$

$$\text{where} \quad \begin{aligned} r^k &= \rho\tilde{z}^k - \rho\tilde{u}^k + A^T H^\ell(A\tilde{x}^k;b)A\tilde{x}^k \\ &+ H^g(\tilde{x}^k)\tilde{x}^k - A^T\nabla\ell(A\tilde{x}^k;b) - \nabla g(\tilde{x}^k). \end{aligned} \tag{5}$$

The inexact ADMM algorithm we propose solves (4) approximately at each iteration.

```
Algorithm 2 Inexact ADMM
input  feature matrix $A$, response $b$, loss function $\ell$, regularization $g$ and $h$, stepsize $\rho$, positive summable sequence $\{\varepsilon^k\}_{k=0}^{\infty}$
  repeat
    find $\tilde{x}^{k+1}$ that solves (4) within tolerance $\varepsilon^k$
    $\tilde{z}^{k+1} = \text{argmin}_z\{h(z) + \frac{\rho}{2}\|\tilde{x}^{k+1} - z + \tilde{u}^k\|_2^2\}$
    $\tilde{u}^{k+1} = \tilde{u}^k + \tilde{x}^{k+1} - \tilde{z}^{k+1}$
  until convergence
output  solution $x_\star$ of problem (1)
```

For a quadratic loss ℓ, when $\sum_{k=0}^{\infty} \varepsilon^k < \infty$ and under various other conditions, if optimization problem (1) has an optimal solution, the $\{\tilde{x}^k\}_{k=0}^{\infty}$ sequence generated by Algorithm 2 converges to the optimal solution of (1) (Eckstein & Bertsekas, 1992; Eckstein & Yao, 2016).

From Boyd et al. (2011), quantity $r_d^{k+1} = \rho(\tilde{z}^k - \tilde{z}^{k+1})$ can be regarded as the dual residual and $r_p^{k+1} = \tilde{x}^{k+1} - \tilde{z}^{k+1}$ can be viewed as the primal residual at iteration $k+1$. This suggests that we can terminate the ADMM iterations when the primal and dual residuals become very small. The primal and dual tolerances can be chosen based on an absolute and relative criterion, such as

$$\begin{aligned}
\|r_p^k\|_2 &\le \epsilon^{\text{abs}} + \epsilon^{\text{rel}}\text{max}\{\|\tilde{x}^k\|_2, \|\tilde{z}^k\|_2\} \\
\|r_d^k\|_2 &\le \epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|\rho\tilde{u}^k\|_2.
\end{aligned}$$

The relative criteria ϵ^{rel} might be 10^{-3} or 10^{-4} in practice. The choice of absolute criteria ϵ^{abs} depends on the scale of the variable values. More details can be found in Boyd et al. (2011).

2.2. Randomized Nyström approximation and PCG

Nyström approximation constructs a low-rank approximation of a symmetric psd matrix H. Let $\Omega \in \mathbb{R}^{d\times s}$ be a test matrix (often, random Gaussian Frangella et al. (2021); Tropp et al. (2017)) with sketch size $s \ge 1$. The Nyström approximation with respect to Ω is given by

$$H\langle\Omega\rangle = (H\Omega)(\Omega^T H\Omega)^\dagger(H\Omega)^T. \tag{6}$$

The Nyström approximation $H\langle\Omega\rangle$ is symmetric, psd, and has rank at most s Lemma A.1. Naive implementation of the

Nyström approximation based on (6) is numerically unstable. Algorithm 4 in Appendix B states a stable procedure to compute a randomized Nyström approximation from Tropp et al. (2017).

Algorithm 4 returns the randomized Nyström approximation of matrix H in the form of an eigendecomposition: $H_{\text{nys}} = U\hat{\Lambda}U^T$. Let $\hat{\lambda}_s$ be the sth eigenvalue. The randomized Nyström preconditioner and its inverse take the form

$$\begin{aligned} P &= \frac{1}{\hat{\lambda}_s + \rho} U(\hat{\Lambda} + \rho I)U^T + (I - UU^T), \\ P^{-1} &= (\hat{\lambda}_s + \rho)U(\hat{\Lambda} + \rho I)^{-1}U^T + (I - UU^T) \end{aligned} \tag{7}$$

(Frangella et al., 2021). In a slight abuse of terminology, we sometimes refer to the sketch size s as the rank of the Nyström preconditioner. We will use the the term sketch size and rank interchangeably throughout the paper. The Nyström preconditioner may be applied to vectors in $O(ds)$ time and only requires $O(ds)$ floating point numbers to store. The details of how to implement PCG with (7) are provided in Appendix B in Algorithm 5. We now provide some background on Nyström PCG and motivation for why we have paired it with ADMM.

Nyström PCG improves on standard CG both in theory and in practice for matrices with a small *effective dimension* (Frangella et al., 2021), which we now define. Given a symmetric psd matrix $H \in \mathbb{R}^{d\times d}$ and regularization $\rho > 0$, the *effective dimension* of H is

$$d_{\text{eff}}(\rho) = \text{tr}(H(H+\rho I)^{-1}). \tag{8}$$

The effective dimension may be viewed as smoothed count of the eigenvalues of H greater than or equal to ρ. We always have $d_{\text{eff}}(\rho) \leq d$, and we expect $d_{\text{eff}}(\rho) \ll d$ whenever H exhibits spectral decay.

In machine learning, most feature matrices naturally exhibit polynomial or exponential spectral decay (Derezinski et al., 2020), thus we expect that $d_{\text{eff}} \ll d$. The randomized Nyström preconditioner in Frangella et al. (2021) exploits the smallness of $d_{\text{eff}}(\rho)$ to build an highly effective preconditioner. Frangella et al. (2021) show that if (7) is constructed with a sketch size $s \gtrsim d_{\text{eff}}(\rho)$, then the condition number of the preconditioned system is constant with high probability. An immediate consequence is that PCG solves the preconditioned system to ϵ-accuracy in $O\left(\log(\frac{1}{\epsilon})\right)$ iterations, independent of the condition number of H.

Observe the Hessian $A^TH^\ell A + H^g$ in the inexact ADMM subproblem (4) is formed from the feature matrix A. Based on the preceding discussion, we expect the Hessian to exhibit spectral decay and for the effective dimension to be small to moderate in size. Hence we should expect Nyström PCG to accelerate the solution of (4) significantly.

2.3. NysADMM

Integrating Nyström PCG with inexact ADMM, we obtain NysADMM, presented in Algorithm 3.

Algorithm 3 NysADMM

```
input  feature matrix $A$, response $b$, loss function $\ell$, regularization $g$ and $h$, stepsize $\rho$, positive summable sequence $\{\varepsilon^k\}_{k=0}^{\infty}$
  $[U, \hat{\Lambda}] = \text{RandNyströmApprox}(A^TH^\ell A + H^g, s)$ {use Algorithm 4 in Appendix B}
  repeat
    use Nyström PCG (Algorithm 5 in Appendix B) to find $\tilde{x}^{k+1}$ that solves (4) within tolerance $\varepsilon^k$
    $\tilde{z}^{k+1} = \text{argmin}_z\{h(z) + \frac{\rho}{2}\|\tilde{x}^{k+1} - z + \tilde{u}^k\|_2^2\}$
    $\tilde{u}^{k+1} = \tilde{u}^k + \tilde{x}^{k+1} - \tilde{z}^{k+1}$
  until convergence
output  solution $x_\star$ of problem (1)
```

Our theory for Algorithm 3, shows that if the sketch size $s \gtrsim d_{\text{eff}}(\rho)$, then with high probability subproblem (4) will be solved to ϵ-accuracy in $O\left(\log(\frac{1}{\epsilon})\right)$ iterations (Corollary 4.2). When the loss ℓ is quadratic and the sequence of tolerances $\{\varepsilon^k\}_{k=0}^{\infty}$ is decreasing with $\sum_{k=0}^{\infty}\varepsilon^k < \infty$, NysADMM is guaranteed to converge as $k \to \infty$ with only a constant number of matvecs per iteration (Theorem 4.3). Table 1 compares the complexity of inexact ADMM with CG vs. NysADMM for K iterations under the hypotheses of Theorem 4.3. NysADMM achieves a significant decrease in runtime over inexact ADMM with CG, as the iteration complexity no longer depends on the condition number κ_2.

Table 1. Complexity comparison, for a quadratic loss with Hessian H. Here T_{mv} is the time to compute a matrix vector product with H, κ_2 is the condition number of H, and ε^k is the precision of the kth subproblem solve (4).

Method	Complexity
Inexact ADMM with CG	$O\left(\sum_{k=1}^{K} T_{\text{mv}}\sqrt{\kappa_2}\log\left(\frac{\kappa_2}{\varepsilon^k}\right)\right)$
NysADMM	$O\left(T_{\text{mv}}d_{\text{eff}}(\rho)\right) +$ $\sum_{k=1}^{K} T_{\text{mv}}\left(4 + \left\lceil 2\log\left(\frac{R}{\varepsilon^k\rho}\right)\right\rceil\right)$

2.4. AdaNysADMM

Two practical problems remain in realizing the success predicted by the theoretical analysis of Table 1. These bounds are achieved by selecting the sketch size to be $d_{\text{eff}}(\rho)$, but the effective dimension is 1) seldom known in practice, and 2) often larger than required to achieve good convergence of NysADMM. Fortunately, a simple adaptive strategy for choosing the sketch size, inspired by Frangella et al. (2021), can achieve the same guarantees as in Table 1. This strategy chooses a tolerance ϵ and doubles the sketch size s until the empirical condition number $\frac{\hat{\lambda}_s+\rho}{\rho}$ satisfies

$$\frac{\hat{\lambda}_s+\rho}{\rho} \leq 1+\epsilon. \tag{9}$$

Theorem 4.4 guarantees that (9) holds when $s \geq d_{\text{eff}}(\rho)$ and that when (9) holds, the true condition number is on the order of $1+\epsilon$ with high probability. We refer to (9) as the empirical condition number as it provides an estimate of the true condition number of the preconditoned system (Theorem 4.4).

Thus, to enjoy the guarantees of Theorem 4.4 in practice, we may employ the adaptive version of NysADMM, which we call AdaNysADMM. We provide the pseudocode for AdaNysADMM in Algorithm 7 in Appendix B. Furthermore, as we use a Gaussian test matrix, it is possible to construct a larger sketch from a smaller one. Hence the total computational work needed by the adaptive strategy is not much larger than if the effective dimension were known in advance. Indeed, AdaNysADMM differs from NysADMM only in the construction of the preconditioner. The dominant cost in forming the precondition is computing the sketch is $H\Omega$, which costs $O(T_{\text{mv}}d_{\text{eff}}(\rho))$. As AdaNysADMM reuses computation, the dominant complexity for constructing the Nyström preconditioner remains $O(T_{\text{mv}}d_{\text{eff}}(\rho))$. Consequently, the overall complexity of AdaNysADMM is the same as NysADMM in Table 1.

3. Applications

Here we discuss various applications that can be reformulated as instances of (1) and solved by Algorithm 3.

3.1. Elastic net

Elastic net generalizes lasso and ridge regression by adding both the ℓ_1 and ℓ_2 penalty to the least squares problem:

$$\text{minimize} \quad \frac{1}{2}\|Ax-b\|_2^2 + \frac{1}{2}(1-\gamma)\|x\|_2^2 + \gamma\|x\|_1 \tag{10}$$

Parameter $\gamma > 0$ interpolates between the ℓ_1 and ℓ_2 penalties. NysADMM applies with $\ell(Ax;b) = \frac{1}{2}\|Ax-b\|_2^2$, $g(x) = \frac{1}{2}(1-\gamma)\|x\|_2^2$, and $h(x) = \gamma\|x\|_1$. The Hessian matrices for ℓ and g are A^TA and $(1-\gamma)I$ respectively.

3.2. Regularized logistic regression

Regularized logistic regression minimizes a logistic loss function together with an ℓ_1 regularizer:

$$\text{minimize} \; -\sum_i \left(b_i(Ax)_i - \log(1+\exp((Ax)_i))\right) + \gamma\|x\|_1 \tag{11}$$

NysADMM applies with $\ell(Ax;b) = -\sum_i \left(b_i(Ax)_i - \log(1+\exp((Ax)_i))\right)$ and $h(x) = \gamma\|x\|_1$. The inexact ADMM update chooses x^{k+1} to minimize a quadratic approximation of the log-likelihood,

$$\text{minimize} \; \frac{1}{2}\sum_i w_i^k(q_i^k - (Ax)_i)^2 + \frac{\rho}{2}\|x - \tilde{z}^k + \tilde{u}^k\|_2^2,$$

where w_i^k and q_i^k depend on the current estimate $\tilde{x}^k$ as

$$\begin{aligned} w_i^k =& \frac{1}{2+\exp(-(A\tilde{x}^k)_i)+\exp((A\tilde{x}^k)_i)} \\ q_i^k =& (A\tilde{x}^k)_i + \frac{b_i - \frac{1}{1+\exp(-(A\tilde{x}^k)_i)}}{w_i^k}. \end{aligned}$$

Therefore, the solution of the x-subproblem can be approximated by solving the linear system

$$(A^T\text{diag}(w^k)A + \rho I)x = \rho\tilde{z}^k - \rho\tilde{u}^k + A^T\text{diag}(w^k)q^k.$$

Here w^k and q^k are the vectors for w_i^k and q_i^k. The Hessian matrix of ℓ is given by $A^T\text{diag}(w^k)A$.

3.3. Support vector machine

To reformulate the SVM problem for solution with NysADMM, consider the dual SVM problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T\text{diag}(b)K\text{diag}(b)x - \mathbf{1}^Tx \\ \text{subject to} & x^Tb = 0 \\ & 0 \leq x \leq C. \end{array} \tag{12}$$

Variable x is the dual variable, b is the label or response, and C is the penalty parameter for misclassification. For linear SVM, $K = A^TA$ where A is a feature matrix; and for nonlinear SVM, K is the corresponding kernel matrix. The SVM problem can be reformulated as (1) by setting $\ell(Ax;b) = \frac{1}{2}x^T\text{diag}(b)K\text{diag}(b)x$, $g(x) = -\mathbf{1}^Tx$, and h is the indicator function for convex constraint set $x^Tb = 0,\ 0 \leq x \leq C$. The Hessian matrix for ℓ is $\text{diag}(b)K\text{diag}(b)$.

4. Convergence analysis

This section provides a convergence analysis for NysADMM. All proofs for the results in this section may be found in Appendix A. First we show Nyström PCG can solve any quadratic problem in a constant number of iterations.

Theorem 4.1. *Let H be a symmetric positive semidefinite matrix, $\rho > 0$ and set $H_\rho = H + \rho I$. Suppose we construct the randomized Nyström preconditioner with sketch size $s \geq 8\left(\sqrt{d_{\text{eff}}(\rho)} + \sqrt{8\log(\frac{16}{\delta})}\right)^2$. Then*

$$\kappa_2(P^{-1/2}H_\rho P^{-1/2}) \leq 8 \tag{13}$$

with probability at least $1-\delta$.

Theorem 4.1 strengthens results in Frangella et al. (2021), which provides sharp expectation bounds on the condition number of the preconditioned system, but gives loose high probability bounds based on Markov's inequality. Our result tightens these bounds, showing that Nyström PCG enjoys an exponentially small failure probability.

As an immediate corollary, we can solve (4) with a few iterations of PCG using the Nyström preconditioner.

Corollary 4.2. *Instate the hypotheses of Theorem 4.1 and let $\tilde{x}_\star$ denote the solution of* (4)*. Then with probability at least $1-\delta$, the iterates $\{x_t\}_{t\geq 1}$ produced by Nyström PCG on problem* (4) *satisfy*

$$\frac{\|x_t - \tilde{x}_\star\|_2}{\|\tilde{x}_\star\|_2} \leq \left(\frac{1}{2}\right)^{t-4}. \tag{14}$$

Thus, after $t \geq \left\lceil \frac{\log\left(\frac{16\|\tilde{x}_\star\|_2}{\epsilon}\right)}{\log(2)} \right\rceil$ iterations,

$$\|x_t - \tilde{x}_\star\|_2 \leq \epsilon. \tag{15}$$

Corollary 4.2 ensures that we can efficiently solve the subproblem to the necessary accuracy at each iteration. This result allows us to prove convergence of NysADMM.

Theorem 4.3. *Consider the problem in* (1) *with quadratic loss $\ell(Ax;b) = \frac{1}{2}\|Ax-b\|_2^2$ and the smooth part g of regularizer r has constant Hessian. Define initial iterates $\tilde{x}^0$, $\tilde{z}^0$ and $\tilde{u}^0 \in \mathbb{R}^d$, stepsize $\rho > 0$, and summable tolerance sequence $\{\varepsilon^k\}_{k=0}^\infty \subset \mathbb{R}_+$. Assume at kth ADMM iteration, the norm of the righthand side of the linear system r^k is bounded by constant R for all k. Construct the Nyström preconditioner with sketch size*

$$s \geq 8\left(\sqrt{d_{\text{eff}}(\rho)} + \sqrt{8\log\left(\frac{16}{\delta}\right)}\right)^2$$

and solve problem (1) *with NysADMM, using $T^k = 4 + \left\lceil 2\log\left(\frac{R}{\varepsilon^k\rho}\right)\right\rceil$ iterations for PCG at the kth ADMM iteration. Then with probability at least $1-\delta$,*

1. *For all $k \geq 0$, each iterate $\tilde{x}^{k+1}$ satisfies*
$$\|\tilde{x}^{k+1} - x^{k+1}\|_2 \leq \varepsilon^k, \tag{16}$$
where x^{k+1} is the exact solution of (4).

2. *As $k \to \infty$, $\{\tilde{x}^k\}_{k=0}^\infty$ converges to a solution of the primal* (1) *and $\{\rho\tilde{u}^k\}_{k=0}^\infty$ converges to a solution of the dual problem of* (1).

Theorem 4.3 establishes convergence of NysADMM for a quadratic loss. The quadratic loss already covers many applications of interest including the lasso, elastic-net, and SVMs. We conjecture that a modification of our argument can show that NysADMM converges linearly for any strongly convex loss, but we leave this extension to future work.

The next result makes rigorous the claims made in Section 2.4: it shows we can determine whether or not we have reached the effective dimension by monitoring the empirical condition number $(\hat{\lambda}_s + \rho)/\rho$.

Theorem 4.4. *Suppose, for some user defined tolerance $\epsilon > 0$, the sketch size satisfies*

$$s \geq 8\left(\sqrt{d_{\text{eff}}\left(\frac{\epsilon\rho}{6}\right)} + \sqrt{8\log\left(\frac{16}{\delta}\right)}\right)^2.$$

Then the empirical condition number of the Nyström preconditioned system $P^{-1/2}H_rP^{-1/2}$ satisfies

$$\frac{\hat{\lambda}_s + \rho}{\rho} \leq 1 + \frac{\epsilon}{42}. \tag{17}$$

Furthermore, with probability at least $1-\delta$,

$$\left|\kappa_2(P^{-1/2}H_\rho P^{-1/2}) - \frac{\hat{\lambda}_s + \rho}{\rho}\right| \leq \epsilon. \tag{18}$$

Theorem 4.4 shows that once the empirical condition number is sufficiently close to 1, so too is the condition number of the preconditioned system. Hence it is possible to reach the effective dimension by doubling the sketch size of the Nyström approximation until the empirical condition number falls below the desired tolerance. Theorem 4.4 ensures the true condition number is close to this empirical estimate with high probability.

Theorem 4.4 also helps explain why sketch sizes much smaller than the effective dimension can succeed in practice. The point is best illustrated by instantiating an explicit parameter selection in Theorem 4.4, which yields the following corollary.

Corollary 4.5. *Instate the hypotheses of Theorem 4.4 with $\epsilon = 100$. Then with a sketch size of $s \gtrsim d_{\text{eff}}(16\rho)$ the following holds*

1. $(\hat{\lambda}_s + \rho)/\rho \leq 1 + \frac{100}{42}$.

2. *With probability at least $1-\delta$,*
$$\left|\kappa_2(P^{-1/2}H_\rho P^{-1/2}) - 1 - \frac{100}{42}\right| \leq 100.$$

Corollary 4.5 shows that for a coarse tolerance of $\epsilon = 100$, a sketch size of $s \gtrsim d_{\text{eff}}(16\rho)$ suffices to ensure that the condition number of $P^{-1/2}H_\rho P^{-1/2}$ is no more than around 100. Two practical observations cement the importance of this corollary. First, $d_{\text{eff}}(16\rho)$ is often significantly smaller than $d_{\text{eff}}(\rho)$, possibly by an order of magnitude or more. Second, with a condition number around 100, PCG is likely to converge very quickly. In fact, for modest condition numbers, PCG is known to converge much faster in practice than the theory would suggest (Trefethen & Bau III, 1997). It is only when the condition number reaches around 10^3, that convergence starts to slow. Thus, Corollary 4.5 helps explain why it is not necessary for the sketch size to equal the effective dimension in order for NysADMM to obtain significant accelerations.

5. Numerical experiments

Table 2. Statistics of experiment datasets.

Name	instances n	features d	nonzero %
STL-10	13000	27648	96.3
CIFAR-10	60000	3073	99.7
CIFAR-10-rf	60000	60000	100.0
smallNorb-rf	24300	30000	100.0
E2006.train	16087	150348	0.8
sector	6412	55197	0.3
p53-rf	16592	20000	100.0
connect-4-rf	16087	30000	100.0
realsim-rf	72309	50000	100.0
rcv1-rf	20242	30000	100.0
cod-rna-rf	59535	60000	100.0

In this section, we evaluate the performance of NysADMM on different large-scale applications: lasso, ℓ_1-regularized logistic regression, and SVM. For each type of problems, we compare NysADMM with popular standard solvers. We run all experiments on a server with 128 Intel Xeon E7-4850 v4 2.10GHz CPU cores and 1056GB. We repeat every numerical experiment ten times and report the mean solution time. We highlight the best-performing method in bold. The tolerance of NysADMM at each iteration is chosen as the geometric mean $\varepsilon^{k+1} = \sqrt{r_p^k r_d^k}$ of the ADMM primal residual r_p and dual residual r_d at the previous iteration, as in (Stellato et al., 2020). See Boyd et al. (2011) for more motivation and details. An alternative is to choose the tolerance sequence as any decaying sequence with respect to the righthand side norm as the number of NysADMM iteration increases, e.g., $\varepsilon^k = \|r^k\|_2/k^\beta$, where β is a predefined factor. These two strategies perform similarly; our experiments use the first strategy.

We choose a sketch size $s = 50$ to compute the Nyström approximation throughout our experiments. Inspired by Theorem 4.4 and Corollary 4.5, even if the sketch size is much smaller than the effective dimension, NysADMM can still achieve significant acceleration in practice.

To support experiments with standard solvers, for each problem class we use the same stopping criterion and other parameter settings as the standard solver. These experiments use datasets with $n > 10,000$ or $d > 10,000$ from LIBSVM (Chang & Lin, 2011), UCI (Dua & Graff, 2017), and OpenML (Vanschoren et al., 2013), with statistics summarized in Table 2. We use a random feature map (Rahimi & Recht, 2008a;b) to generate features for the data sets CIFAR-10, smallnorb, realsim, rcv1, and cod-rna, which increases both predictive performance and problem dimension.

5.1. Lasso

This subsection demonstrates the performance of NysADMM to solve the standard lasso problem (2). Here we compare NysADMM with three standard lasso solvers, SSNAL (Li et al., 2018), mfIPM (Fountoulakis et al., 2014), and glmnet (Friedman et al., 2010). SSNAL is a Newton method based solver; mfIPM is an interior point method based solver and glmnet is a coordinate descent based solver. In practice, these three solvers and NysADMM rely on different stopping criteria. In order to make a fair comparison, in our experiments, the accuracy of a solution x for (2) is measured by the following relative Karush–Kuhn–Tucker (KKT) residual (Li et al., 2018):

$$\eta = \frac{\|x - \operatorname{prox}_{\gamma\|\cdot\|_1}(x - A^T(Ax - b))\|}{1 + \|x\| + \|Ax - b\|}. \tag{19}$$

For a given tolerance ϵ, we stop the tested algorithms when $\eta < \epsilon$. Note that stopping criterion (19) is rather strong: if $\eta \le 10^{-2}$ for NysADMM, then the primal and dual gaps for ADMM are $\lesssim 10^{-4}$, which suffices for most applications. Indeed, for many machine learning problems, lower bounds on the statistical performance of the estimator (Loh, 2017) imply an unavoidable level of statistical error that is greater than this optimization error for most applications. Optimizing the objective beyond the level of statistical error (Agarwal et al., 2012; Loh & Wainwright, 2015) does not improve generalization. For standard lasso experiments, we fix the regularization parameter at $\gamma = 1$.

Table 3. Results for low precision lasso experiment.

Task	Time for $\epsilon = 10^{-1}$ (s)			
	NysADMM	mfIPM	SSNAL	glmnet
STL-10	**165**	573	467	278
CIFAR-10-rf	**251**	655	692	391
smallNorb-rf	**219**	552	515	293
E2006.train	**313**	875	903	554
sector	**235**	678	608	396
realsim-rf	**193**	–	765	292
rcv1-rf	**226**	563	595	273
cod-rna-rf	**208**	976	865	324

Table 3 and Table 4 show results for lasso experiments. The average solution time for NysADMM, mfIPM, SSNAL, and

Table 4. Results for high precision lasso experiment.

Task	Time for $\epsilon = 10^{-2}$ (s)			
	NysADMM	mfIPM	SSNAL	glmnet
STL-10	**406**	812	656	831
CIFAR-10-rf	**715**	1317	1126	1169
smallNorb-rf	**596**	896	768	732
E2006.train	1657	1965	**1446**	2135
sector	957	1066	**875**	1124
realsim-rf	**732**	–	1035	922
rcv1-rf	**593**	853	715	736
cod-rna-rf	**715**	1409	1167	997

glmnet with $\epsilon = 10^{-1}, 10^{-2}$ on different tasks are provided. Here mfIPM fails to solve the realsim-rf instance since it requires $n < d$. For precision of $\epsilon = 10^{-1}$, NysADMM is faster than all other solvers and at least 3 times faster than both mfIPM and SSNAL. For precision of $\epsilon = 10^{-2}$, NysADMM is still faster than all other solvers for all instances except E2006.train and sector. The results are fair since both SSNAL and mfIPM are second-order solvers and can reach high precision. NysADMM and glmnet are first-order solvers; they reach low precision quickly, but improve accuracy more slowly than a second order method. In practice, for large-scale machine learning problems, a low precision solution usually suffices, as decreasing optimization error beyond the statistical noise in the problem does not improve generalization. Further, our algorithm achieves bigger improvements on dense datasets compared with sparse datasets, as the factors of the Nyström approximation are dense even for sparse problems. To further illustrate the results, we vary the value of ϵ from 1.0 to 10^{-3} on STL-10 task and plot the average solution time for four methods in Figure 1. We can see NysADMM is as least as fast as other solvers when $\epsilon > 10^{-3}$, and often twice as fast for many practical values of ϵ.

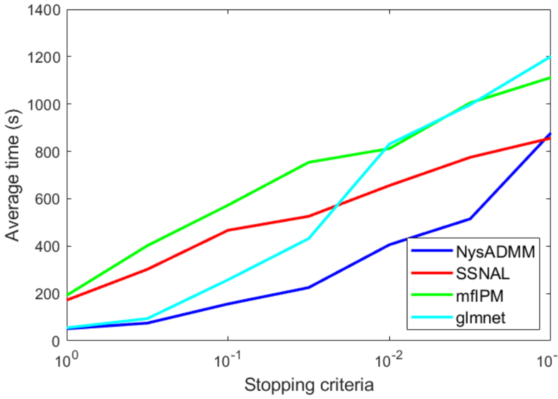

Figure 1. Solution times for varying tolerance ϵ on STL-10.

5.2. ℓ_1-regularized logistic regression

This subsection demonstrates the performance of NysADMM on ℓ_1-regularized logistic regression, (11) from Section 3.2. We test the method on binary classification problems using the same random feature map as in Section 5.1.

The ℓ_1-regularized logistic regression experiments compare NysADMM with the SAGA algorithm, a stochastic average gradient like algorithm (Defazio et al., 2014) implemented in sklearn, and the accelerated proximal gradient (APG) algorithm (Beck & Teboulle, 2009; Nesterov, 2013; O'Donoghue & Candes, 2015). For the purpose of fair comparison, all the algorithms are stopped when the maximum relative change in the problem variable (that is, the regression coefficients) $\frac{\|x_k - x_{k+1}\|_\infty}{\|x_k\|_\infty}$ is less than the tolerance. The tolerance is set to 10^{-3}; other settings match the default settings of the sklearn logistic regression solver.

An overview of ℓ_1-regularized logistic regression experiment results are provided in Table 5. NysADMM uniformly out performs SAGA, solving each problem at least twice as fast. Similarly, NysADMM is at least twice as fast as APG on all datasets except STL-10, where it performs comparably. In the cases of p53-rf and connect-4-rf, NysADMM runs significantly faster than its competitors, being four times faster than SAGA and three times faster than APG. These large performance gains are due to the size of the problem instances and their conditioning. From (Defazio et al., 2014), the convergence speed of SAGA depends on the problem instance size and condition number. Our test cases have large instance sizes and condition numbers, which lead to slow convergence of SAGA. The situation with APG is similar. Indeed, although ADMM and proximal gradient methods generally have the same $O(1/t)$-convergence rate (Beck & Teboulle, 2009; He & Yuan, 2012), NysADMM is less sensitive ill-conditioning than APG.

Table 5. Results for ℓ_1-regularized logistic regression experiment.

Task	NysADMM (s)	SAGA (s)	APG (s)
STL-10	3012	6083	**2635**
CIFAR-10-rf	**7884**	21256	17292
p53-rf	**528**	2116	1880
connect-4-rf	**866**	4781	7365
smallnorb-rf	**1808**	6381	4408
rcv1-rf	**1237**	3988	2759
con-rna-rf	**7528**	21513	16361

5.3. Support vector machine

This subsection demonstrates the performance of NysADMM on kernel SVM problem for binary classification, (12) from Section 3.3. The SVM experiments compare NysADMM with the LIBSVM solver (Chang & Lin, 2011). LIBSVM uses sequential minimal optimization (SMO) to solve the dual SVM problem. We use the same stopping criteria as the LIBSVM solver, which stops the NysADMM method when the ADMM dual gap reaches 10^{-4} level. All SVM experiments use the RBF kernel. Table 6 shows the results of SVM experiments. On these problems, NysADMM is at least 3 times faster (and up to 58 times faster) than the LIBSVM solver. Consider problem formulation (12), with the RBF kernel. The Gram matrix $\mathrm{diag}(b)K\mathrm{diag}(b)$ is dense and approximately low rank: exactly the setting in which NysADMM should be expected to perform well. In constrast, the SMO-type decomposition in LIBSVM solver works better for sparse problems, as it updates only two variables at each iteration.

Table 6. Results of SVM experiment.

Task	NysADMM time (s)	LIBSVM time (s)
STL-10	**208**	11573
CIFAR-10	**1636**	8563
p53-rf	**291**	919
connect-4-rf	**7073**	42762
realsim-rf	**17045**	52397
rcv1-rf	**564**	32848
cod-rna-rf	**4942**	36791

6. Conclusion

In this paper, we have developed a scalable new algorithm, NysADMM, that combines inexact ADMM and the randomized low-rank Nyström approximation to accelerate composite convex optimization. We show that NysADMM exhibits strong benefits both in theory and in practice. Our theory shows that when the Nyström preconditioner is constructed with an appropriate rank, NysADMM requires only a constant number of matvecs to solve the ADMM subproblem. We have also provided an adaptive strategy for selecting the rank that possesses a similar computational profile to the non-adaptive algorithm, and allows us to realize the theoretical benefits in practice. Further, numerical results demonstrate that NysADMM is as least twice as fast as standard methods on large dense lasso, regularized logistic regression, and kernalized SVM problems. More broadly, this paper shows the promise of recent advances in RandNLA to provide practical accelerations for important large-scale optimization algorithms.

References

Agarwal, A., Negahban, S., and Wainwright, M. J. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

Alaoui, A. and Mahoney, M. W. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, 2015.

Bach, F. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, 2013.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

Chowdhuri, A., London, P., Avron, H., and Drineas, P. Speeding up linear programming using randomized linear algebra. In *Advances in Neural Information Processing Systems*, 2020.

Cohen, M. B., Nelson, J., and Woodruff, D. P. Optimal approximate matrix product in terms of stable rank. In *43rd International Colloquium on Automata, Languages, and Programming*, 2016.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 2014.

Derezinski, M., Liang, F. T., Liao, Z., and Mahoney, M. W. Precise expressions for random projections: Low-rank approximation and randomized Newton. In *Advances in Neural Information Processing Systems*, 2020.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Eckstein, J. and Bertsekas, D. P. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

Eckstein, J. and Yao, W. Approximate versions of the alternating direction method of multipliers. *Optimization Online*, 2016.

Fountoulakis, K., Gondzio, J., and Zhlobich, P. Matrix-free interior point method for compressed sensing problems. *Mathematical Programming Computation*, 6(1): 1–31, 2014.

Frangella, Z., Tropp, J. A., and Udell, M. Randomized Nyström preconditioning. *arXiv preprint arXiv:2110.02820*, 2021.

Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Gower, R. M., Kovalev, D., Lieder, F., and Richtárik, P. RSN: randomized subspace Newton. In *Advances in Neural Information Processing Systems*, 2019.

He, B. and Yuan, X. On the o(1/n) convergence rate of the Douglas–Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

Lacotte, J. and Pilanci, M. Effective dimension adaptive sketching methods for faster regularized least-squares optimization. In *Advances in Neural Information Processing Systems*, 2020.

Lacotte, J. and Pilanci, M. Fast convex quadratic optimization solvers with adaptive sketching-based preconditioners. *arXiv preprint arXiv:2104.14101*, 2021.

Li, X., Sun, D., and Toh, K.-C. A highly efficient semismooth newton augmented lagrangian method for solving lasso problems. *SIAM Journal on Optimization*, 28(1): 433–458, 2018.

Loh, P.-L. On lower bounds for statistical learning theory. *Entropy*, 19(11):617, 2017.

Loh, P.-L. and Wainwright, M. J. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16 (19):559–616, 2015.

Martinsson, P.-G. and Tropp, J. A. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

Meng, X., Saunders, M. A., and Mahoney, M. W. LSRN: A parallel iterative solver for strongly over-or underdetermined systems. *SIAM Journal on Scientific Computing*, 36(2):C95–C118, 2014.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1):125–161, 2013.

O'Donoghue, B. and Candes, E. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

Pilanci, M. and Wainwright, M. J. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017.

Platt, J. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*, 1998.

Rahimi, A. and Recht, B. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, 2008a.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008b.

Rokhlin, V. and Tygert, M. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36): 13212–13217, 2008.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12 (4):637–672, 2020.

Trefethen, L. N. and Bau III, D. *Numerical linear algebra*, volume 50. SIAM, 1997.

Tropp, J. A., Yurtsever, A., Udell, M., and Cevher, V. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. In *Advances in Neural Information Processing Systems*, 2017.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Yue, H., Yang, Q., Wang, X., and Yuan, X. Implementing the alternating direction method of multipliers for big datasets: A case study of least absolute shrinkage and selection operator. *SIAM Journal on Scientific Computing*, 40(5):A3121–A3156, 2018.

A. Proofs of main results

In this section we give the proofs for the main results of the paper: Theorem 4.1, Theorem 4.3, and Theorem 4.4.

A.1. Preliminaries

We start by recalling some useful background information and technical results that are useful for proving the main theorems. In order to obtain the exponentially small failure probabilities in Theorem 4.1 and Theorem 4.4 we take a different approach from the one in Frangella et al. (2021). The proofs are based on regularized Schur complements and approximate matrix multiplication. Our arguments are inspired by the techniques used to establish statistical guarantees for approximate kernel ridge regression via column sampling schemes (Alaoui & Mahoney, 2015; Bach, 2013).

A.1.1. Nyström Approximation: Properties

We start by recalling some important properties of the Nyström approximation (4). We shall also need the regularized Nyström approximation. Recall that $\Omega \in \mathbb{R}^{d\times s}$ denotes the test matrix from which we construct the Nyström approximation. Given $\sigma > 0$, the regularized Nyström approximation with respect to Ω is defined as

$$H\langle\Omega\rangle_\sigma = (H\Omega)(\Omega^T H\Omega + \sigma I)^{-1}(H\Omega)^T. \tag{20}$$

Furthermore, let $H = V\Lambda V^T$ be the eigendecomposition of H and define $D_\sigma = H(H+\sigma I)^{-1} = \Lambda(\Lambda+\sigma I)^{-1}$. We shall see below that D_σ plays a crucial role in the analysis. The following lemmas are well known in the literature and summarize the properties of the Nyström and regularized Nyström approximation. Lemma A.1 may be found in Frangella et al. (2021) and Lemma A.2 in Alaoui & Mahoney (2015).

Lemma A.1. *Let $H\langle\Omega\rangle$ be a Nyström approximation of a symmetric psd matrix H. Then*

1. *The approximation $H\langle\Omega\rangle$ is psd and has rank at most s.*
2. *The approximation $H\langle\Omega\rangle$ depends only on* $\mathrm{range}(\Omega)$.
3. *In the Loewner order, $H\langle\Omega\rangle \preceq H$.*
4. *In particular, the eigenvalues satisfy $\lambda_j(H\langle\Omega\rangle) \leq \lambda_j(H)$ for each $1 \leq j \leq d$.*

Lemma A.2. *Let H be a symmetric psd matrix, $\sigma > 0$. Define $E = H - H\langle\Omega\rangle$ and $E_\sigma = H - H\langle\Omega\rangle_\sigma$. Then the following hold.*

1. $H\langle\Omega\rangle_\sigma \preceq H\langle\Omega\rangle \preceq H$.
2. $0 \preceq E \preceq E_\sigma$.
3. *If $\|D_\sigma^{1/2}V^T(\frac{1}{s}\Omega\Omega^T)VD_\sigma^{1/2} - D_\sigma\| \leq \eta < 1$, then*

$$0 \preceq E_\sigma \preceq \frac{\sigma}{1-\eta}I. \tag{21}$$

Lemma A.2 relates $H\langle\Omega\rangle_\sigma$ to $H\langle\Omega\rangle$ and H. In particular, item 2 implies that $\|E\| \leq \|E_\sigma\|$, so controlling E_σ controls E. Item 3 shows that E_σ can be controlled by the spectral norm of the matrix

$$D_\sigma^{1/2}V^T\frac{1}{s}\Omega\Omega^T VD_\sigma^{1/2} - D_\sigma. \tag{22}$$

The spectral norm of (22) can be bounded by observing

$$\mathbb{E}\left[D_\sigma^{1/2}V^T\frac{1}{s}\Omega\Omega^T VD_\sigma^{1/2}\right] = \tag{23}$$

$$D_\sigma^{1/2}V^T\mathbb{E}\left[\frac{1}{s}\Omega\Omega^T\right]VD_\sigma^{1/2} = \tag{24}$$

$$D_\sigma^{1/2}V^T VD_\sigma^{1/2} = D_\sigma. \tag{25}$$

Thus, $D_\sigma^{1/2}V^T\frac{1}{s}\Omega\Omega^T VD_\sigma^{1/2}$ is an unbiased estimator of D_σ, and may be viewed as approximating the product of the matrices $D_\sigma^{1/2}V^T$ and $VD_\sigma^{1/2}$. Hence results from randomized linear algebra can bound the spectral norm of this difference. In particular, it suffices to take a sketch size that scales with the effective dimension, using results on approximate matrix multiplication in terms of stable rank (Cohen et al., 2016).

A.1.2. Approximate matrix multiplication in terms of the effective dimension

The condition in item 3 of Lemma A.2 follows immediately from theorem 1 of Cohen et al. (2016). Unfortunately, the analysis in that paper does not yield explicit constants. Instead we use a special case of their results due to Lacotte & Pilanci (2021) that provides explicit constants. Theorem A.3 simplifies theorem 5.2 in Lacotte & Pilanci (2021).

Theorem A.3. *Let $\Psi \in \mathbb{R}^{s\times d}$ be a matrix with i.i.d. $N(0,\frac{1}{s})$ entries. Given $\delta > 0$, and $\tau \in (0,1)$ it holds with probability at least $1-\delta$ that*

$$\sup_{v\in\mathbb{S}^{d-1}} \langle v, (D_\sigma^{1/2}V^T\Psi^T\Psi VD_\sigma^{1/2} - D_\sigma)v\rangle \leq \tau + 2\sqrt{\tau}, \tag{26}$$

$$\inf_{v\in\mathbb{S}^{d-1}} \langle v, (D_\sigma^{1/2}V^T\Psi^T\Psi VD_\sigma^{1/2} - D_\sigma)v\rangle \geq \tau - 2\sqrt{\tau}, \tag{27}$$

provided $s \geq \frac{\left(\sqrt{d_{\text{eff}}(\sigma)}+\sqrt{8\log(16/\delta)}\right)^2}{\tau}$.

Setting $\Psi = \frac{1}{\sqrt{s}}\Omega^T$, where $\Omega \in \mathbb{R}^{d\times s}$ has i.i.d. $N(0,1)$ entries, Theorem A.3 yields the following corollary.

Corollary A.4. *Let $\Omega \in \mathbb{R}^{d\times s}$ be a matrix with i.i.d. $N(0,1)$ entries. Given $\delta > 0$, and $\tau \in (0,1)$ it holds with probability at least $1-\delta$ that*

$$\left\|D_\sigma^{1/2}V^T\frac{1}{s}\Omega\Omega^T VD_\sigma^{1/2} - D_\sigma\right\| \leq \tau + 2\sqrt{\tau} \tag{28}$$

provided $s \geq \frac{\left(\sqrt{d_{\text{eff}}(\rho))}+\sqrt{8\log(16/\delta)}\right)^2}{\tau}$.

A.1.3. Condition number of Nyström preconditoned linear system

The following result is a simpler version of proposition 5.2 in Frangella et al. (2021).

Proposition A.5. *Let $\hat{H} = U\hat{\Lambda}U^T$ be any rank-s Nyström approximation, with sth largest eigenvalue $\hat{\lambda}_s$, and let $E = H - \hat{H}$ be the approximation error. Construct the Nyström preconditioner P as in (7). Then the condition number of the preconditioned matrix $P^{-1/2}H_\rho P^{-1/2}$ satisfies*

$$\kappa_2(P^{-1/2}H_\rho P^{-1/2}) \leq \frac{\hat{\lambda}_s + \rho + \|E\|}{\rho}. \tag{29}$$

Proposition A.5 bounds the condition condition number of the Nyström preconditioned linear system in terms of $\hat{\lambda}_s, \rho$ and the approximation error $\|E\|$. We would like to emphasize that the bound in Proposition A.5 is deterministic.

A.2. Proofs of Theorem 4.1 and Corollary 4.2

We start with two lemmas from which Theorem 4.1 follows easily. The first lemma and its proof appear in Frangella et al. (2021).

Lemma A.6. *Let $H \in \mathbb{S}_n^+(\mathbb{R})$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Let $\rho > 0$ be regularization parameter, and define the effective dimension as in (8). Then the following statement holds.*

Fix $\gamma > 0$. If $j \geq (1+\gamma^{-1})d_{\text{eff}}(\rho)$, then $\lambda_j \leq \gamma\rho$.

Lemma A.7. *Let $\epsilon > 0$ and $E = H - H\langle\Omega\rangle$. Suppose we construct a randomized Nyström approximation from a standard Gaussian random matrix Ω with sketch size $s \geq 8\left(\sqrt{d_{\text{eff}}(\epsilon)} + \sqrt{8\log(\frac{16}{\delta})}\right)^2$. Then the event*

$$\mathcal{E} = \{\|E\| \leq 6\epsilon\}, \tag{30}$$

holds with probability at least $1-\delta$.

Proof. Let $\Omega_s = \frac{1}{\sqrt{s}}\Omega$ and observe that $H\langle\Omega_s\rangle = H\langle\Omega\rangle$. Now the conditions of Corollary A.4 are satisfied with $\sigma = \epsilon$ and $\tau = 8$. Consequently with probability at least $1-\delta$,

$$\left\|D_\epsilon^{1/2}V^T\frac{1}{s}\Omega\Omega^T VD_\epsilon^{1/2} - D_\epsilon\right\| \leq \frac{1}{8} + \frac{\sqrt{2}}{2}.$$

Hence applying Lemma A.2 with $\sigma = \epsilon$ and $\eta = \frac{1}{8} + \frac{\sqrt{2}}{2}$, we obtain

$$\left\|H - H\langle\Omega_s\rangle_\epsilon\right\| \leq 6\epsilon,$$

with probability at least $1-\delta$. Recalling our initial observation, we conclude the desired result. □

A.2.1. Proof of Theorem 4.1

Proof. As $s \geq 8\left(\sqrt{d_{\text{eff}}(\rho)} + \sqrt{8\log(\frac{16}{\delta})}\right)^2$ we have that $\|E\| \leq 6\rho$ with probability at least $1-\delta$ by Lemma A.7. Furthermore, $\hat{\lambda}_s \leq \frac{\rho}{7}$ by item 3 of Lemma A.1 and Lemma A.6 with $\gamma = 1/7$. Combining this with Proposition A.5, we conclude with probability at least $1-\delta$,

$$\begin{aligned}\kappa_2(P^{-1/2}H_\rho P^{-1/2}) &\leq \frac{\hat{\lambda}_s + \rho + \|E\|}{\rho}\\ &\leq 1 + 6 + \frac{1}{7} \leq 8\end{aligned}$$

as desired. □

A.2.2. Proof of Corollary 4.2

Proof. Let $A = P^{-1/2}H_\rho P^{-1/2}$ and condition on the event that $\kappa_2(A) \leq 8$, which holds with probability at least $1-\delta$. The standard theory for convergence of CG (Trefethen & Bau III, 1997) guarantees after t iterations that,

$$\frac{\|x_t - \tilde{x}_\star\|_A}{\|\tilde{x}_\star\|_A} \leq 2\left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^t \tag{31}$$

where $\|x\|_A = x^TAx$. Theorem 4.1 guarantees that the Nyströmpreconditioned matrix satisfies $\kappa_2(A) \leq 8$, so the above display may be majorized as

$$\frac{\|x_t - \tilde{x}_\star\|_A}{\|\tilde{x}_\star\|_A} \leq \left(\frac{1}{2}\right)^{t-1}. \tag{32}$$

Now, from the elementary inequality

$$\lambda_d(A)\|x\|_2 \leq \|x\|_A \leq \lambda_1(A)\|x\|_2, \tag{33}$$

we conclude

$$\frac{\|x_t - \tilde{x}_\star\|_2}{\|\tilde{x}_\star\|_2} \leq \kappa_2(A)\left(\frac{1}{2}\right)^{t-1} \leq \left(\frac{1}{2}\right)^{t-4}. \tag{34}$$

To obtain the claimed result, multiply both sides by $\|\tilde{x}_\star\|_2$ and solve $\|\tilde{x}_\star\|_2\left(\frac{1}{2}\right)^{t-4} = \epsilon$ for t. □

A.3. Proof of Theorem 4.3

This proof is a natural consequence of the following theorem from Eckstein & Bertsekas (1992).

Theorem A.8. *Consider a convex optimization problem in the primal form (P),* minimize $f(x) + h(Mx)$*, where* $x \in \mathbb{R}^d$*,* $M \in \mathbb{R}^{m\times d}$ *has full column rank. Pick any* y^0*,* $z^0 \in \mathbb{R}^m$*, and* $\rho > 0$*, and summable sequences*

$$\begin{aligned}&\{\varepsilon^k\}_{k=0}^\infty \subseteq [0,\infty),\ \sum_{k=0}^\infty \varepsilon^k < \infty,\\ &\{\nu^k\}_{k=0}^\infty \subseteq [0,\infty),\ \sum_{k=0}^\infty \nu^k < \infty,\\ &\{\lambda^k\}_{k=0}^\infty \subseteq (0,2),\ 0 < \inf\lambda^k \leq \sup\lambda^k < 2.\end{aligned}$$

The dual problem (D) of primal problem (P) is

$$\text{maximize}_{y\in\mathbb{R}^m} \ -(f^*(-M^Ty)+g^*(y)).$$

Suppose the primal and dual ADMM iterates $\{x^k\}_{k=0}^{\infty}$, $\{z^k\}_{k=0}^{\infty}$, and $\{y^k\}_{k=0}^{\infty}$ satisfy the update equations to within errors given by conform, for all k to

$$\begin{aligned}
&\Bigg\|x^{k+1}-\text{argmin}_x\big\{f(x)+\langle y^k, Mx\rangle\\
&+\frac{1}{2}\rho\|Mx-z^k\|_2^2\big\}\Bigg\|_2 \le \varepsilon^k,\\
&\Bigg\|z^{k+1}-\text{argmin}_z\big\{h(z)-\langle y^k, z\rangle\\
&+\frac{1}{2}\rho\|\lambda^kMx^{k+1}-z+(1-\lambda^k)z^k\|_2^2\big\}\Bigg\|_2 \le \nu^k,\\
&y^{k+1}=y^k+\rho(\lambda^kMx^{k+1}+(1-\lambda^k)z^k-z^{k+1}).
\end{aligned} \tag{35}$$

Then if (P) has a Kuhn-Tucker pair, $\{x^k\}$ converges to a solution of (P) and $\{y^k\}$ converges to a solution of (D).

A.3.1. Proof of Theorem 4.3

Proof. Consider optimization problem (1) and the associated NysADMM algorithm Algorithm 3. Suppose $\{\tilde{x}^k\}_{k=0}^{\infty}$, $\{\tilde{z}^k\}_{k=0}^{\infty}$, and $\{\tilde{u}^k\}_{k=0}^{\infty}$ are generated by NysADMM iterations. Since $\ell(Ax,b)$ is quadratic with respect to x and the smooth part g of regularizer r has constant Hessian, the x-subproblem of (1) is exactly the linear system (4).

Let x^{k+1} be the exact solution for the x-subproblem at iteration k. For all $k\ge 0$, NysADMM iterate $\tilde{x}^{k+1}$ satisfies $\|\tilde{x}^{k+1}-x^{k+1}\|_2\le\varepsilon^k$. Let $M=I$, $\nu^k=0$, $\lambda^k=1$, $y^k=\rho\tilde{u}^k$ for all k, and $f(x)=\ell(Ax,b)+g(x)$. By Theorem A.8, $\{\tilde{x}^k\}_{k=0}^{\infty}$, $\{\tilde{z}^k\}_{k=0}^{\infty}$, and $\{\rho\tilde{u}^k\}_{k=0}^{\infty}$ satisfy condition (35). Therefore, if optimization problem (1) has a Kuhn-Tucker pair, $\{\tilde{x}^k\}$ converges to a solution of (1) and $\{\rho\tilde{u}^k\}$ converges to a solution of the dual problem of (1).

Next, we derive the bound for the number of Nyström PCG iterations T^k required at NysADMM iteration k. Note that in this case the Hessians of ℓ and g are constant. We only need to sketch once for the constant linear system matrix $A^TH^{\ell}(A\tilde{x}^k;b)A+H^g(\tilde{x}^k)$ and can reuse the sketch for all NysADMM iterations. Since the Nyström preconditioner is constructed with sketch size $s\ge 8\left(\sqrt{d_{\text{eff}}(\rho)}+\sqrt{8\log(\frac{16}{\delta})}\right)^2$, by Corollary 4.2, with probability at least $1-\delta$, after

$$T^k\ge\left\lceil\frac{\log\left(\frac{16\|x^{k+1}\|_2}{\varepsilon^k}\right)}{\log(2)}\right\rceil$$

Nyström PCG iterations, we have $\|\tilde{x}^{k+1}-x^{k+1}\|_2\le\varepsilon^k$. Recall the righthand side of linear system (4) r^k. The exact solution for the x-subproblem x^{k+1} at iteration k satisfies $\|x^{k+1}\|_2\le\frac{\|r^k\|_2}{\rho}$. We have

$$\left\lceil\frac{\log\left(\frac{16\|x^{k+1}\|_2}{\varepsilon^k}\right)}{\log(2)}\right\rceil\le\left\lceil\frac{\log\left(\frac{16\|r^k\|_2}{\varepsilon^k\rho}\right)}{\log(2)}\right\rceil.$$

Further, by assumption, as $\|r^k\|_2$ is bounded by a constant R for all k, we have

$$\left\lceil\frac{\log\left(\frac{16\|r^k\|_2}{\varepsilon^k\rho}\right)}{\log(2)}\right\rceil\le 4+\left\lceil\frac{\log\left(\frac{R}{\varepsilon^k\rho}\right)}{\log(2)}\right\rceil\le 4+\left\lceil 2\log\left(\frac{R}{\varepsilon^k\rho}\right)\right\rceil.$$

This gives the bound for the number of Nyström PCG iterations T^k required at NysADMM iteration k □

A.4. Proof of Theorem 4.4

Proof. By hypothesis we have $s>8d_{\text{eff}}(\frac{\epsilon\rho}{6})$, so Lemma A.6 with $\gamma=7$ yields

$$\hat{\lambda}_s\le\lambda_s\le\frac{1}{7}\frac{\epsilon\rho}{6}=\frac{\epsilon\rho}{42},$$

Thus,

$$\frac{\hat{\lambda}_s + \rho}{\rho} \leq 1 + \frac{\epsilon}{42}.$$

This gives the first statement. For the second statement we use our hypothesis on s to apply Lemma A.7 with tolerance $\epsilon\rho/6$. From this we conclude $\|E\| \leq \epsilon\rho$ with probability at least $1-\delta$. Combining this with Proposition A.5 yields

$$\kappa_2(P^{-1/2}H_\rho P^{-1/2}) - \frac{\hat{\lambda}_s + \rho}{\rho} \leq \epsilon,$$

with probability at least $1-\delta$. On the other hand, condition numbers always satisfy

$$\kappa_2(P^{-1/2}H_\rho P^{-1/2}) \geq 1.$$

Combining this with our upper bound on $\hat{\lambda}_s$ gives

$$\begin{aligned} \kappa_2(P^{-1/2}H_\rho P^{-1/2}) - \frac{\hat{\lambda}_s + \rho}{\rho} &\geq 1 - (1 + \epsilon/42) \\ &= -\epsilon/42. \end{aligned}$$

Hence with probability at least $1-\delta$

$$\left|\kappa_2(P^{-1/2}H_\rho P^{-1/2}) - \frac{\hat{\lambda}_s + \rho}{\rho}\right| \leq \epsilon.$$

□

B. Randomized Nyström approximation and Nyström PCG

In this section we give the algorithms from Frangella et al. (2021) for the randomized Nyström approximation and Nyström PCG.

Algorithm 4 Randomized Nyström Approximation

```
input  psd matrix $H \in \mathbb{S}_d^+(\mathbb{R})$, sketch size $s$
  $\Omega = \text{randn}(d, s)$ {Gaussian test matrix}
  $\Omega = \text{qr}(\Omega, 0)$ {thin QR decomposition}
  $Y = H\Omega$ {$s$ matvecs with $H$}
  $\nu = \text{eps}(\text{norm}(Y, 2))$ {compute shift}
  $Y_\nu = Y + \nu\Omega$ {add shift for stability}
  $C = \text{chol}(\Omega^T Y_\nu)$ {Cholesky decomposition}
  $B = Y_\nu / C$ {triangular solve}
  $[U, \Sigma, \sim] = \text{svd}(B, 0)$ {thin SVD}
  $\hat{\Lambda} = \max\{0, \Sigma^2 - \nu I\}$ {remove shift, compute eigs}
output  Nyström approximation $\hat{H}_{\text{nys}} = U\hat{\Lambda}U^T$
```

C. AdaNysADMM

In this section we give the adaptive algorithm for computing the randomized Nyström approximation adopted from Frangella et al. (2021). The adaptive algorithm has the benefit of reusing computation, in particular, we do not need to compute the sketch Y from scratch. We simply add onto the sketch that we have already computed. We also give the pseudo-code for AdaNysADMM that uses Algorithm 6 to compute the Nyström preconditioner.

Algorithm 5 Nyström PCG

```
input  psd matrix $H$, righthand side $r$, initial guess $x_0$, regularization parameter $\rho$, sketch size $s$, tolerance $\varepsilon$
  $[U, \hat{\Lambda}] = \text{RandomizedNyströmApproximation}(H, s)$
  $w_0 = r - (H + \rho I)x_0$
  $y_0 = P^{-1}w_0$
  $p_0 = y_0$
  while $\|w\|_2 > \varepsilon$ do
    $v = (H + \rho I)p_0$
    $\alpha = (w_0^T y_0)/(p_0^T v)$
    $x = x_0 + \alpha p_0$
    $w = w_0 - \alpha v$
    $y = P^{-1}w$
    $\beta = (w^T y)/(w_0^T y_0)$
    $x_0 \leftarrow x$, $w_0 \leftarrow w$, $p_0 \leftarrow y + \beta p_0$, $y_0 \leftarrow y$
  output  approximate solution $\hat{x}$
```

Algorithm 6 AdaptiveRandNysAppx

```
input  symmetric psd matrix $H$, initial rank $s_0$, tolerance Tol
  $Y = [\ ], \Omega = [\ ]$, and $(\hat{\lambda}_s + \rho)/\rho = \text{Inf}$
  $m = s_0$
  while $(\hat{\lambda}_s + \rho)/\rho > \text{Tol}$ do
    generate Gaussian test matrix $\Omega_0 \in \mathbb{R}^{n \times m}$
    $[\Omega_0, \sim] = \text{qr}(\Omega_0, 0)$
    $Y_0 = H\Omega_0$
    $\Omega = [\Omega\ \ \Omega_0]$ and $Y = [Y\ \ Y_0]$
    $\nu = \sqrt{n}\,\text{eps}(\text{norm}(Y, 2))$
    $Y_\nu = Y + \nu\Omega,$
    $C = \text{chol}(\Omega^T Y_\nu)$
    $B = Y_\nu / C$
    compute $[U, \Sigma, \sim] = \text{svd}(B, 0)$
    $\hat{\Lambda} = \max\{0, \Sigma^2 - \nu I\}$ {remove shift}
    compute $(\hat{\lambda}_s + \rho)/\rho$
    $m \leftarrow s_0$, $s_0 \leftarrow 2s_0$ {double rank if tolerance is not met}
    if $s_0 > s_{\text{max}}$ then
      $s_0 = s_0 - m$ {when $s_0 > s_{\text{max}}$, reset to $s_0 = s_{\text{max}}$}
      $m = s_{\text{max}} - s_0$
      generate Gaussian test matrix $\Omega_0 \in \mathbb{R}^{n \times m}$
      $[\Omega_0, \sim] = \text{qr}(\Omega_0, 0)$
      $Y_0 = H\Omega_0$
      $\Omega = [\Omega\ \ \Omega_0]$ and $Y = [Y\ \ Y_0]$
      $\nu = \sqrt{n}\,\text{eps}(\text{norm}(Y, 2))$ {compute final approximation and break}
      $Y_\nu = Y + \nu\Omega,$
      $C = \text{chol}(\Omega^T Y_\nu)$
      $B = Y_\nu / C$
      compute $[U, \Sigma, \sim] = \text{svd}(B, 0)$
      $\hat{\Lambda} = \max\{0, \Sigma^2 - \nu I\}$
      break
    end if
  end while
output  Nyström approximation $(U, \hat{\Lambda})$
```

Algorithm 7 AdaNysADMM

```
input feature matrix $A$, response $b$, loss function $\ell$, regularization $g$ and $h$, stepsize $\rho$, positive summable sequence $\{\varepsilon^k\}_{k=0}^{\infty}$
  $[U, \hat{\Lambda}] =$ AdaptiveRandNysAppx(
  A$^T H^\ell A + H^g, s)$ {use Algorithm 6}
  repeat
    find $\tilde{x}^{k+1}$ that solves (4) within tolerance $\varepsilon^k$ by Nyström PCG
    $\tilde{z}^{k+1} = \text{argmin}_z \{h(z) + \frac{\rho}{2}\|\tilde{x}^{k+1} - z + \tilde{u}^k\|_2^2\}$
    $\tilde{u}^{k+1} = \tilde{u}^k + x^{k+1} - \tilde{z}^{k+1}$
  until convergence
```