Complexity analysis for AdaRBFGS: a primitive for methods between first and second order

Joel Castellon
Supervisor: Dr. Sebastian Stich and Prof. Martin Jaggi
Machine Learning and Optimization Laboratory, IC School, EPFL

***Abstract*—There is a fundamental trade-off in first and second order methods between complexity of the updates (e.g. time and memory) and surface information an algorithm can incorporate per iteration. AdaRBFGS, a sub-routine for such methods, iteratively approximates matrices (such as the Hessian) and incorporates a dimension reducing sketch while adaptively accelerating its convergence. Our work aims to shed some light on the convergence behavior of AdaRBFGS which was proposed as an heuristic in [1].**

I. Introduction

Matrix inversion algorithms are a fundamental sub-routine in optimization methods. Such sub-routines play an important role in the trade-off between incorporating surface information and time/memory efficiency. Which is particularly important in second order methods. In this work we consider adaRBFGS as a primitive for minimizing $f(x)$ that is smooth and convex (additional assumptions such as separability permit further specializations as in [2]). Perhaps, the most well-known example is Newton's method with updates of the form $x_{k+1} = x_k - \alpha_k H_k^{-1} \nabla f(x_k)$ (where α_k is the step size, x_k newton's method iterates and H_k the Hessian of f at iteration k). In such updates we are required to calculate the inverse of the Hessian (at each step) in order to incorporate second-order curvature information. While Newton's method has an excellent convergence rate (e.g. quadratic), the computational burden of calculating the inverse Hessian or even storing it. This amounts to $O(n^2)$ for storage and $O(n^3)$ flops to calculate such update, with no prior structure or information about the Hessian. Making it in-adequate for many applications. On the other hand, a gradient descent update has the form $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$ with $O(n)$ per iteration complexity both in time and memory. Then, the question is whether we can get some of the best of both worlds.

The motivation for this work comes from studying a primitive proposed by Gower et al. in [1]. In Gower's work, the adaRBFGS method is proposed in an heuristic manner and stems from a more general setting that has solid theoretical ground. The authors compare, empirically, adaRBFGS against other baselines for inverting positive definite matrices and adaRBFGS outperforms these by far. Yet, there is still little understanding of the convergence properties of the rate for adaRBFGS as well as other more specific properties (such as dependence on the dimension of the sketch matrix, see Subsection VI-E). Such fine grained analysis of the rate for adaRBFGS is the subject of this work and also a suggested direction for further reasearch by Gower et al. in [1].

II. Outline

We start this report by describing the general framework that is at the origin of adaRBFGS in Section IV as background. The proposed method by Gower et al. is interesting, first, because of the generalization (c.f. Subsection IV-A) and the connexion this framework makes with previously well known family of methods such as quasi-newton and simultaneous Kaczmarz just to name a few. In Section V we mention some results and properties of adaRBFGS that will be useful for our complexity analysis. We will, then, show our results

with respect to the rate complexity (Section VI) of the rate via numerical experiments and theoretical justification. Finally, we make suggestions for future research and open problems we encountered in Section VII.

III. Notation

We briefly outline some notation and assumptions that will be used in this report. We follow closely the notation established by Gower et al. in [1] to make the parallel between such work and ours easier to follow. First, the matrix that we aim to invert is $A \in \mathbb{R}^{n\times n}$ which is assumed to be symmetric and positive definite (extensions for non-symmetric and non-psd can be found in [1], [4]). Then, $X_k \in \mathbb{R}^{n\times n}$ are the sequence of iterates that aim to approximate A. We often refer to the corresponding Cholesky factors $L_k \in \mathbb{R}^{n\times n}$ (where $X_k = L_k L_k^T$, such decomposition stems from (5) when we discuss sampling (as in Subsection VI-C). Next, $S \in \mathbb{R}^{n\times q}$ is the sketch matrix to be sampled at each iteration with support denoted by $\mathcal{S} \in \mathbb{R}^{n\times n}$ (details on such distribution are given in Subsection V-B). Also, let (1) be the weighted Frobenius norm.

$$\begin{aligned} \|X\|^2_{F(W^{-1})} &= Tr(X^T W^{-1} X W^{-1}) \\ &= \left\|W^{-1/2} X W^{-1/2}\right\|^2_F \end{aligned} \tag{1}$$

Where $W \in \mathbb{R}^{n\times n}$ is a parameter of the algorithm as presented in [1] (this is set to A^{-1} for adaRBFGS). And, $\|.\|_F$ is the standard Frobenius norm.

IV. RBFGS

AdaRBFGS emerges from a framework proposed by Gower et al. in [1], [2] to which we refer as *Stochastic Iterative Matrix Inversion* (see Algorithm 1). The idea is to generalize quasi-newton (qN) updates, which are the primitives (matrix updates for approximating the inverse Hessian) in qN-methods. For example, take the following unconstrained optimization problem (with f smooth and convex). $\min_{x\in\mathbb{R}^n} f(x)$. The quasi-newton iteration being,

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k) \tag{2}$$

Now, any quasi-newton matrix B_k (approximation of the Hessian) must satisfy the following constraint.

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k) \tag{3}$$

This last expression is normally referred as the *secant equations*.

A. General setting and geometric interpretation

The first idea of Gower et al. in [1], [2] is to reduce the dimension of the system in (3) while maintaining low rank updates to matrices that aim to approximate the inverse Hessian. AdaRBFGS as developed in [1] only considers the following version of the unconstrained optimization problem $f(x) = \frac{1}{2}x^T A x - b^T x + c$. Furthermore, the secant equations become $AX = I$. Therefore, we have A^{-1} as our target inverse Hessian, and the sequence of iterates to approximate A^{-1} [1] are given by:

$$\begin{aligned} X_{k+1} = \text{argmin}_{X\in\mathbb{R}^{n\times n}} \{ &\frac{1}{2}\|X - X_k\|^2_{F(W^{-1})} \\ &\text{subject to } S^T A X = S^T \} \end{aligned} \tag{4}$$

$W \in \mathbb{R}^{n\times n}$ being a parameter of the algorithm, which is set to A^{-1} for adaRBFGS. Where $S \in \mathbb{R}^{n\times q}$ and $q \ll n$. We call S the sketch matrix that will reduce the dimension of the system of equations. For example, in the extreme case one can take $S = e_i$. For such choice we have to solve a system with 1 row and n columns (e.g. $e_i^T A X = e_i^T$). Another basic case is when we set $S = I$, this yields the original secant equations for our problem (e.g. $AX = I$). In its general form, S can be considered a random matrix, we draw S either from a fixed or varying distribution (see Subsection IV-C). Such reduction in complexity for the constraints is, of course, at the expense of introducing many spurious solutions which is why we aim to take small rank updates. Gower et al. call this first formulation for the X_k updates the *sketch-and-project* point of view. Note that the sketch-and-project point of view is a convex quadratic problem with affine constraints. Therefore, strong duality

[1] yet the authors of [1] claim it can be adapted to approximate arbitrary matrices

holds and the following dual update rule can be derived (see [1])

$$X_{k+1} = \text{argmin}_{X,Y}\{\frac{1}{2}\left\|X - A^{-1}\right\|^2_{F(W^{-1})} \quad \text{subject to } X = X_k + WA^TSY^T\} \tag{5}$$

This last expression is referred to as the *constrain-and-approximate* point of view. There are two interesting points about this dual formulation. First, the latter type of update corresponds to the *Approximate Inverse Preconditioning (AIP)* family of methods, while the former kind of update corresponds to qN updates. Then, this framework shows a previously unknown connection between qN and AIP methods. The second important point about the constrain-and-approximate rule is a geometrical interpretation which gives a more intuitive understanding of what the method does. What the constrain-and-approximate update rule says is that at each iterate X_k we draw a random hyperplane crossing such point (over the distribution of S). Next, take the orthogonal projection of the target A^{-1} in such random hyperplane as the iterate. Hence, the algorithm consists of drawing random hyperplanes crossing the iterates and projecting the objective which is A^{-1}.

Algorithm 1 Stochastic Iterative Matrix Inversion

Input:

$A \in \mathbb{R}^{n\times n}$ ▷ a symmetric matrix to be inverted
$\mathcal{D}$ ▷ distribution over random matrices
$W \in \mathbb{R}^{n\times n}$ ▷ symmetric pos. def. $W = A^{-1}$ for RBFGS
$X_0 \in \mathbb{R}^{n\times n}$ ▷ symmetric initial iterate

1: **for** k=0,1,2... **do**
2: Sample an independent copy $S \sim \mathcal{D}$
3: Compute $\Lambda = S(S^TAWAS)^{-1}S^T$
4: $\Theta = \Lambda AW$, $M_k = X_kA - I$

$$X_{k+1} = X_k - M_k\Theta - (M_k\Theta)^T + \Theta^T(AX_kA - A)\Theta$$

5: **end for**

Output: last iterate X_k

B. RBFGS

In the framework described above there are three main parameters (see Algorithm 1). Namely, the distribution of random matrices $\mathcal{D}$ where $S \sim \mathcal{D}$, the weights matrix W and assumptions on A (e.g. symmetric or positive definite). It turns out that most well-known quasi newton methods can be obtained by varying these these three parameters (see [1] for a full description). The most relevant result, for us, in this category is obtaining the Broyden-Fletcher-Goldfarb-Shanno (BFGS) by setting $W = A^{-1}$, S as a deterministic vector and A semi positive definite. The next step is to let S be a random matrix (of full column rank) that we draw a fixed distribution for which we obtain RBFGS. RBFGS is a randomized block variant of the BFGS method as it can be seen as taking a block of qN-directions per iteration. This can be seen, for example, in the derivation of the qN updates in [6] where S is the scaled direction of the update (e.g. a vector). [2]
The main result, that will concern us, regarding the rate of this non-adaptive algorithm (e.g. RBFGS) is Thm. 6.1 in [1] which is the analogous of Thm. 9.1 we outline in this report.

Theorem 6.2 (Gower et al. 2016)**.** *After applying k iterations of Algorithm 1 (Assuming A and X_0 are symmetric) we have:*

$$\mathbf{E}\left[\left\|X_k - A^{-1}\right\|^2_{F(W^{-1})}\right] \leq \rho^k \left\|X_0 - A^{-1}\right\|^2_{F(W^{-1})} \tag{6}$$

where $\rho = 1 - \lambda_{min}(W^{1/2}\mathbf{E}[Z]W^{1/2})$,
and $0 \leq 1 - \mathbf{E}[q]/n \leq \rho \leq 1$.

Where the expectation is taken over the distribution of S. Note that in Thm. 6.2, the rate depends on the quantity $1 - \rho = \lambda_{\min}(W^{1/2}\mathbf{E}[Z]W^{1/2})$. Where $W^{1/2}ZW^{1/2}$ is a projection matrix (as shown in Lemma 6.1 in [1]) that projects the iterates X_k into a sketch of the column space of A (e.g. $W^{1/2}A^TS$, see Lemma 6.1 in [1] for details). In fact, the form that $1 - \rho$ has will be recurrent in our analysis as there

[2] More generally, the authors of [6] set S as a random matrix (this time without a rigorous derivation) and proceed with this setup in the rest of [6]

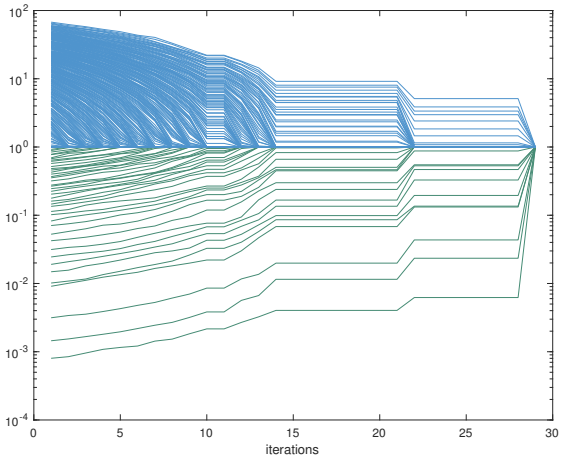

Fig. 1: Each trajectory corresponds to the eigenvalue of a fixed rank (e.g. smallest, second smallest, ..., largest) across iterations. This plot corresponds to the spectrum of a $AX_k \in \mathbb{R}^{300\times 300}$ with A being symmetric and positive definite. This experiment puts in evidence the interlacing spectrum property of adaRBFGS (column variant) on which we elaborate in Thm. 2.3

is a adaptive counterpart (in Thm.9.1). Furthermore, our results rely on bounds that will be obtained for such adaptive rate (Section VI) similarly to Thm.6.2 above.

C. Adaptive RBFGS

Next, we outline the intuition for adaRBFGS as derived in [1]. First, the main convergence result in [1] is Thm. 6.2 which outlines a rate (see Section VI) in expected norm of the error. We assume that the sketch matrix sampled at each iteration S_i comes from a bigger matrix (its support) which we call $\mathcal{S}$ (conditions on Section V). In section 7.1 of [1] the problem of determining optimal sampling probabilities is formulated as a semi definite program. Solving such SDP, however, can be even costlier than inverting A itself as we iterate (support given by experiments by the authors in [1]). Nonetheless, the formula for the optimal probabilities obtained by the SDP (formula (62) in [1]) is insightful in the sense that it suggests that optimal sampling is its dependence on A^{-1}. However, since we do not have A^{-1} we can use X_k as a proxy. This is, in fact, the intuition behind the heuristic which states that sampling gets better (this being in close relation to the rate, see Section VI) as the estimate X_k is closer to the target A^{-1}. Now, in order to get a specific form for the support $\mathcal{S}$, an upper bound for the rate is derived in [2]. Such upper bound is optimal by setting $\mathcal{S} = A^{-T}W^{-1/2} = A^{-1/2}$. This means that if X_k approximates A^{-1} then we should sample $S = S_i$ (the sketch) at each iteration from the columns of L_k (e.g. the Cholesky factor of X_k). In fact an update rule that uses only these factors was derived in [6] in the context of limited memory preconditioners for BFGS. This latter update form is the one adaRBFGS uses in [1]

Next, we present two variants for the varying distribution where S is sampled from (e.g. $\mathcal{D}_k$). A first such distribution is the *adaRBFGS gauss* (baseline results in Fig. 2). In adaRBFGS gauss we set $S_i = X_k G$ where $G \in \mathbb{R}^{n\times q}$ is a random matrix of standard gaussian i.i.d entries.

The second variant of adaRBFGS we present is the *adaRBFGS col*. This version consists of sampling S as a column sub-matrix S_i from $L_k = [S_1 \dots S_r]$ (where $L_k L_k^T = X_k$) with sampling probabilities p_i (as defined in Subsection VI-C). More formally, $S = L_k I\colon C_i = S_i$ where C_i is the index set corresponding to column sub-matrix S_i.

D. Adaptive RBFGS as a primitive

The RBFGS update rule (which is the same for adaRBFGS but with varying $\mathcal{D}$, see Section V) has a well-known form in the BFGS literature, and a consequence of this form is that it can be composed in a larger setting to get methods with well-known properties in the quasi Newton family. A concrete example is the work in [3] where the authors combine RBFGS to approximate sub-sampled Hessians, obtain memory efficient variants (in the spirit of L-BFGS) and even combine it with variance reduction techniques for the gradient [18]. In the more general setting, adaRBFGS can be used as a primitive for preconditioning. In fact, when the output has the guarantee to be positive definite it can be used in the design of variable metric optimization methods (see [15], [17]).

Algorithm 2 AdaRBFGS

Input:

A ▷ a symmetric positive definite matrix to be inverted

$\mathcal{D}$ ▷ distribution over random matrices with n rows

$L_0 \in \mathbb{R}^{n\times n}$ ▷ pick invertible initial iterate

for k=0,1,2... **do**

Sample an independent copy $\mathcal{S} \sim \mathcal{D}$ ▷ $\mathcal{D}$ may also depend on k as in the column variant

Compute $S = L_k\mathcal{S}$ ▷ S is sampled adaptively as it depends on k

Compute $R_k = (\mathcal{S}^T A\mathcal{S})^{-1/2}$

$$L_{k+1} = L_k + SR_k((\mathcal{S}^T\mathcal{S})^{-1/2}\mathcal{S}^T - R_k^T S^T A L_k)$$

▷ Update the factor

end for

Output: $X_k = L_k L_k^T$

E. Numerical experiments

Next we present results that we reproduced from [1] by re-running one of the main experiments of such work. The difference is, however, that we changed the implementation provided in [5] to perform the sampling for the column variant of adaRBFGS as stated in [1]. The original implementation performed uniform sampling with respect to the column index, while we now perform *convenient sampling* as in (12) and (13). The results are qualitatively the same as in [1]. We plot the error (e.g. distance to A^{-1}) against the baseline methods as observed in Fig. 2. Observe that adaRBFGS outperforms largely other well established methods. The experiments in Fig. 2 were performed in a 1000×1000 symetric positive semi definite matrix. The two methods we use as benchmark are Minimal residual (a method with global convergence) and Newton-Schulz which is basically Newton's method for calculating matrix inverse. As rates for adaRBFGS are the main subject of study for this

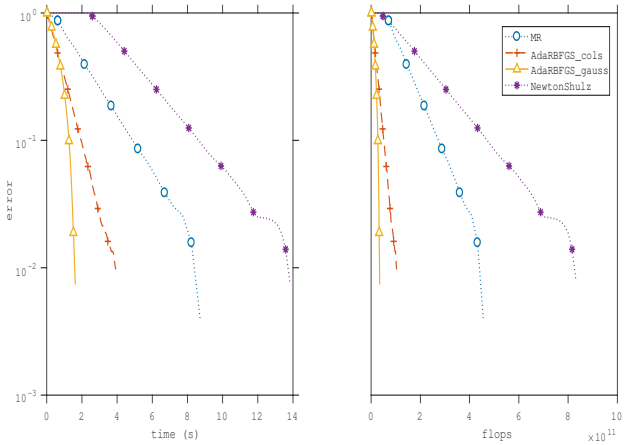

Fig. 2: Empirical comparison of Ada-RBFGS variants against two standard methods for matrix inversion (Minimal residual and Newton-Schulz) for a 1000×1000 symmetric matrix.

project we have extended the implementation [3] to track the rate, bounds we propose and related metrics.

The Matrix used in the experiments is $A = BB^T$ where $B \in \mathbb{R}^{n\times n}$ is obtained by sampling standard Gaussian i.i.d entries. Typical condition numbers for such matrices are in the order of $1e^{10}$. Smaller matrices (e.g. 100,500,700 dimensions) had, correspondingly, condition numbers that were orders of magnitude smaller. We observed that adaRBFGS starts outperforming Minimal Residual and Newton Schulz for larger matrices. For matrices beyond 1000 dimensions MR and Newton Schulz would not even converge.

V. Properties of RBFGS

In this section we outline the main properties that will be useful for our analysis in Section VI . A part from the main results regarding the rate, we outline some more fundamental properties such as the update rule for the X_k. In fact, iterates in the form of (5) is the reason why adaRBFGS can be used as part of unconstrained convex function minimization methods and known results in BFGS literature can be re-derived for this setting (as outlined in Subsection IV-D). So, even if the expression

[3] code available at https://github.com/epfml/adarbfgs-joel-castellon

for the update below does not seem to be very insightful, it is at the core of the properties we mention here.

A. The update rule

The update expression can be obtained from the constrain-and-approximate point of view (e.g. Subsection IV-A) by replacing the constraints in the objective and enforcing symmetry with an additional constraint (detailed derivation in [1]). We then obtain:

$$X_{k+1} = S(S^TAS)^{-1}S^T + (I - S(S^TAS)^{-1}S^TA)X_k(I - AS(S^TAS)^{-1}S^T) \tag{5}$$

One can see from equation (5) that the update preservers symmetry and positive definitiveness (factorize X_k and its left/right factors are symmetric, see [1] for a full proof). These two latter properties are important since we guarantee all iterates are non-singular, and symmetry is required in the results we use in Subsections VI-D and VI-E (analysis of convergence and dependence of sketch dimension). Furthermore, this form also makes possible the Cholesky factorization and thus factored update derived in [6].

B. One-step progress and complete discrete sampling

The starting point for an analysis of convergence is Thm. 6.2 in [1] (for which the one-step version is Thm.9.1). Then, an important question is: what conditions should a sketch matrix satisfy in order to get a result like Thm. 6.2 in [1]. Actually, in order to have such a result the adaptive setting, the authors give a characterization for the support $\mathcal{S}$ where the S_i (sketch matrices) are sampled from. Let $S = S_i \in \mathbb{R}^{n\times q_i}$ with probability $p_i > 0$ for $i \in [r]$ where S_i has full column rank. And S is defined as a complete discrete sampling when $\mathcal{S} = [S_1 \dots S_r] \in \mathbb{R}^{n\times n}$ has full row rank.
Now, under the condition that the matrix we sample at each iteration is a complete discrete sampling we have:

Theorem 9.1 (Gower et al. 2016)**.** *After one step of AdaRBFGS method we have*

$$\mathbf{E}\Big[\, \|X_{k+1} - A^{-1}\|^2_{F(A)} \,|X_k\Big] \leq \rho_k \, \|X_k - A^{-1}\|^2_{F(A)} \tag{6}$$

where $1 - \rho_k = \lambda_{min}(A^{-1/2}\mathbf{E}\big[Z|X_k\big]A^{-1/2})$

Given that the rate ρ_k does not have an immediate interpretation, the authors in [1] propose a convenient sampling as in (12) which yields the following upper bound $\rho_k \leq 1 - \frac{\lambda_{\min}(AX_k)}{Tr(AX_k)}$.
Note that such bound depends on the spectrum of A preconditioned by X_k. Using the same form, A preconditioned with X_k, we obtain in Section VI a tighter upper bound for the rate and even a lower bound.

VI. Results

A. Interlacing spectrum

As mentioned in Section V, equation (5) has some important consequences, among these is the interlacing spectrum (Thm. 2.3). Such update rule is a known result in the BFGS literature and can be derived under different circumstances. One such example comes from the equivalent formulation given by conjugate gradient with exact line search (as shown in [8]). Moreover, a simple corollary of the interlacing spectrum is that the condition number is non-increasing (which we use as motivation in the derivation of our proposed sampling in Thm. 2) is a result independently found by Fletcher in [9]. Next, we state the interlacing spectrum theorem as applied to our setting.

Theorem 2.3 (Interlacing spectrum Gratton 2011)**.** *Let $\sigma_1, \dots \sigma_n$ real positive be the eigenvalues of AX_k arranged in non-decreasing order. And, q being the dimension of the sketch matrix $S \in \mathbb{R}^{n\times q}$. Then, the eigenvalues $\mu_1, \dots, \mu_n$ of AX_{k+1} can be arranged so that*

$$\begin{cases} \sigma_j \leq \mu_j \leq \sigma_{j+q} \text{ for } j \in \{1, \dots, n-q\} \\ \mu_j = 1 \text{ for } j \in \{n-q+1, \dots, n\} \end{cases} \tag{7}$$

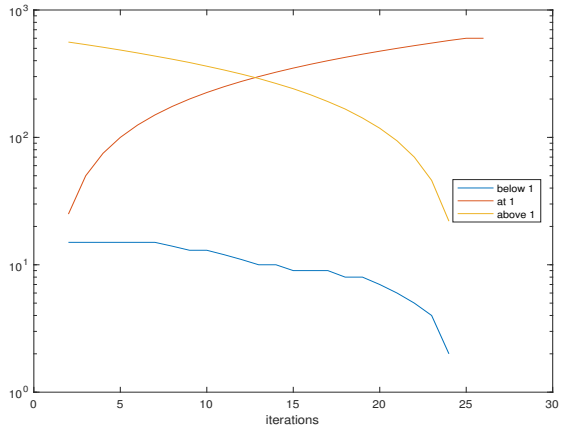

Fig. 3: Eigenvalue count for AX_k as a function of iterations for adaRBFGS (column variant).

It was first observed that adaRBFGS satisfies the interlacing spectrum property by Stich et al. in [15]. The proof is the same as that in [6] where the theorem we present is derived in a slightly different setting, namely, limited-memory preconditioners for least-squares problems. Note, however, that the proof basically relies on the form of the update rule for X_k (details in [6]) which can be factorized in two orthogonal components as presented in the following expression (part of the derivation in [6]).

$$A^{1/2}X_{k+1}A^{1/2} = \tag{8}$$
$$\begin{bmatrix} A^{1/2}W, & A^{1/2}\underline{W} \end{bmatrix} \begin{bmatrix} I_q & 0 \\ 0 & V\Lambda V^T \end{bmatrix} \begin{bmatrix} W^T A^{1/2} \\ \underline{W}^T A^{1/2} \end{bmatrix}$$

Where W and $\underline{W}$ are A orthogonal matrices, and $W = S(S^TAS)^{-1/2}$ (e.g. a term inside the factors that multiply X_k when updating to X_{k+1}). We observe in equation (8) that the first component I_q has all 1 eigenvalues. Therefore, at each iteration there are q eigenvalues that are projected to 1. Now, the second component

$$V\Lambda V^T = Q^T A^{1/2} X_k A^{1/2} Q \tag{9}$$

(where $Q = A^{1/2}\underline{W}$) is a compression of the eigenvalues of the previous iterate (e.g. AX_k) which yields the interlacing property (see Corollary 4.3.16 in [7]).

We can see interlacing spectrum property (c.f. Thm. 2.3) experimentally in Fig. 1. To confirm this Fig. 1 we clearly see that the eigenvalue trajectories (each trajectory corresponds to a rank) do not cross each other. Furthermore, these are monotonically decreasing for eigenvalues that start above 1 (in blue) and monotonically increasing for those starting below 1 (in green). Lastly, note the groups of lines converging (e.g. being projected) at 1 every few other iterations. Note how this is also described in Thm. 2.3 and in Proposition 2.8 of [6] (we will use this last proposition in Subsections VI-D and VI-E)

We also plot (in Fig. 3) the count of eigenvalues below, above and at 1 with thresholds defined by a tolerance band of $1e^{-2}$ by default. Note in Fig. 3 both eigenvalues above and below one decrease in a monotone manner while those at 1 increase strictly with the number of iterations which is evidence for the interlacing property. Although the spectrum for our input matrix A always crosses 1, when this is not the case we can always re-scale A to have spectrum such that $0 < \lambda_{\text{min}}(A) \leq 1$ and $1 \leq \lambda_{\text{max}}(A)$ (see [15], [6] without changing the system of equations to solve).

One important detail we observe in our experiments is the dependence on the number of iterations until convergence with respect to the initial maximum eigenvalue (e.g. $\lambda_{max}(A)$). To put this in evidence, we show the eigenvalue histogram for adaRBFGS with a tolerance (relative error with respect to the initial error) of $1e^{-2}$ in Fig. 4 and $1e^{-5}$ in Fig. 5. Observe that for a higher error threshold ($1e^{-2}$) the histogram has a tail to the right side which indicates ill conditioning of AX_k. And, as mentioned before, the number of iterations to 'loose' such tail is proportional to $\lambda_{max}(A)$. On the other side, for lower error threshold ($1e^{-5}$) we see that the eigenvalue distribution is centered around 1 with little variation (e.g. $AX_k \approx I$). The authors of adaRBFGS (c.f. [1]) make an intuitive guess on the asymptotic behavior of the spectrum. Namely, the idea of sampling from the Cholesky factor L_k (see Section IV-C) comes from the expectation of an optimized upper bound on the rate. Namely, for the following upper bound

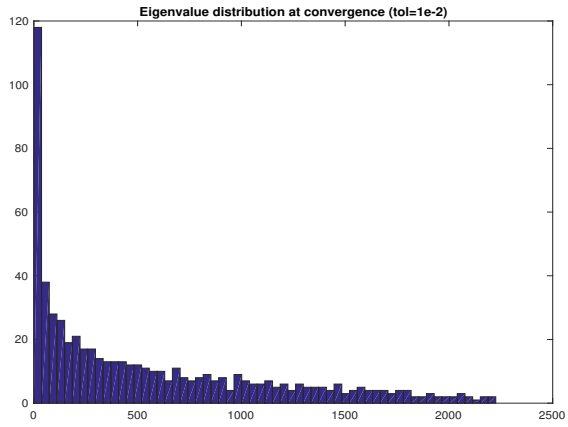

Fig. 4: Eigenvalue histogram for AX_k after convergence, for the column variant of adaRBFGS, with tolerance $1e{-}2$ for the relative error.

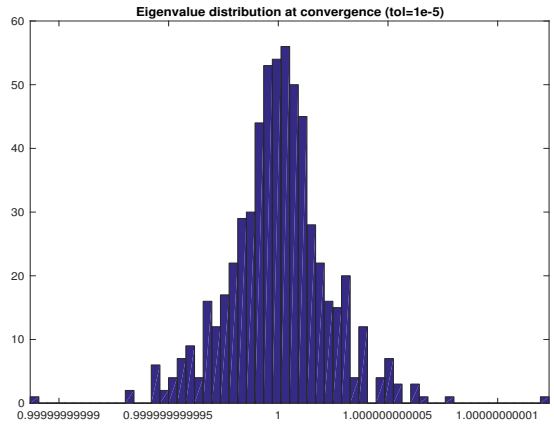

Fig. 5: Eigenvalue histogram after convergence, for the column variant of adaRBFGS, with tolerance $1e-5$ for the relative error.

derived by Gower et al in [2].

$$\rho_k \leq 1 - \frac{\lambda_{min}(AX_k)}{Tr(AX_k)} \tag{10}$$

Gower et al. make the conjecture that $1 - \frac{\lambda_{min}(AX_k)}{Tr(AX_k)} \rightarrow 1 - \frac{1}{n}$. Hence, with this in mind, the design of adaRBFGS aims to make AX_k well-conditioned in the limit. Yet, Gower et al. only give an intuitive argument (based on Lemma 7.2 in [1]) on why this should happen and leave a more detailed analysis for future research.

Once the interlacing property has been identified, a first observation is that the rate depends in the condition number (as it often does in BFGS or conjugate gradient methods). In fact, Thm.2.3 implies that the matrix AX_k gets better conditioned (e.g. smaller gap between largest and smallest eigenvalues) and we would thus hope to converge to $X_k = A^{-1}$ (as $AX_k \approx I$). We conclude this section by exploring the interlacing spectrum property in the adaRBFGS gauss variant in Fig. 6. We note that we have have a similar behavior as that for the spectrum of the column variant. However, note that those eigenvalues above 1 get projected to 1 at a much lower rate as in the column variant case. This experiment gives us some intuition in that AX_k also gets better conditioned for the gaussian variant, which also hints why adaRBFGS in general has great performance against baselines in Fig. 2.

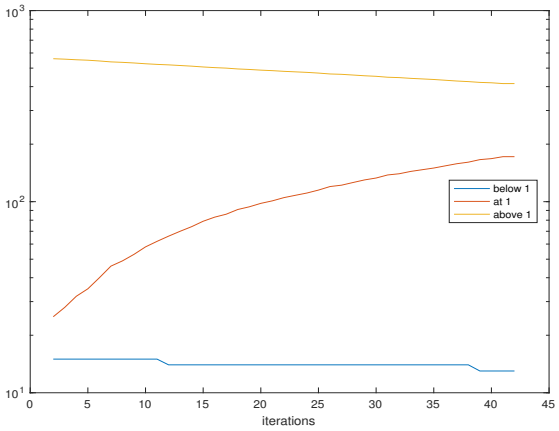

Fig. 6: Eigenvalue count as a function of iterations for adaRBFGS for the gaussian variant.

However, the results we present in Sections VI-B and VI-C have only been derived for the column variant. Hence, a theoretical analysis for gaussian or other adaptive distributions is left open for extension in future research.

B. Lower bound on adaptive rate

From Thm. 2.3, we have derived the following lower bound

Theorem 1 (Lower bound on adaptive rate for AdaRBFGS)**.** *Let ρ_k be the adaptive rate as in Thm. 9.1. Then, we can obtain the following lower bound:*

$$\rho_k \geq 1 - \frac{tr(D^2)}{tr(A^{-1/2}(L_k^T)^{-1}(L_k)^{-1}A^{-1/2})} \qquad (11)$$
$$= \rho_{lower}$$

Where
$D = Diag(\sqrt{p_1}(S_1^T A S_1)^{-1/2}, \ldots, \sqrt{p_r}(S_r^T A S_r)^{-1/2})$

Proof. This comes, in fact, from a corollary of the interlace theorem for symmetric matrices as derived in [12], [11]. Namely, $\lambda_{min}(Q)tr(R) \leq tr(QR)$ for Q, R real symmetric and positive semi-definite. Then, set $Q = A^{-1/2}\mathbf{E}\big[Z|X_k\big]A^{-1/2} = A^{1/2}L_k D^2 L_k^T A^{1/2}$ (such factorization for $\mathbf{E}\big[Z|X_k\big]$ is detailed in Proposition 7.1 of [1]) and $R = A^{-1/2}(L_k^T)^{-1}(L_k)^{-1}A^{-1/2}$. Note that $tr(R)$ is positive as the factors L_k we maintain are positive definite thanks to the form in equation (5). □

Compare ρ_{lower} for both adaRBFGS variants in Figures 7 and 8. Note that this lower bound bound is especially useful for experiments using the sampling strategy proposed in [1] (equation (12)) because the upper bound (e.g. ρ_{upper}) is rather loose near convergence (this can be seen in Fig. 7). Thus ρ_{upper} is not really a good indicator of the potential performance of the true rate (it is almost 2 orders of magnitude off the true rate). Hence the lower bound, in this case, is a better indicator of the potential performance of the true rate.
Finally, we mention that we have also tried another rate lower bound which is a direct consequence of the interlacing theorem (see [12]), namely Schur's theorem $diag(B) \prec \lambda(B)$ where B is the matrix in the definition of ρ_k. However, such lower bound did not present a quantitative improvement in our experiments, so we keep the first one.

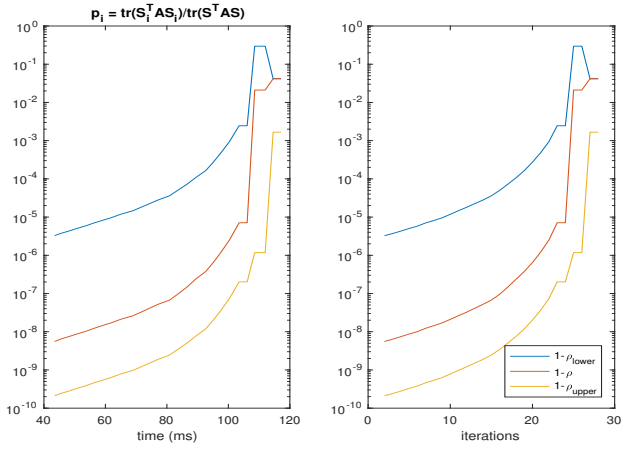

Fig. 7: $1 - \rho_k = \lambda_{\min}(A^{-1/2}\mathbf{E}\big[Z|X_k\big]A^{-1/2})$ and lower/upper bounds (as in equations (11) and (10) respectively). These plots correspond to the column variant of adaRBFGS with p_i as in (12).

C. A more concise convenient sampling and non-increasing upper bound

The reason why Gower et al. choose the sampling probabilities as

$$p_i = \frac{tr(S_i^T A S_i)}{tr(\mathcal{S}^T A \mathcal{S})} \qquad (12)$$

is that such sampling yields an optimal upper bound. Namely, $1 - \frac{\lambda_{min}(AX_k)}{tr(AX_k)} \to 1 - \frac{1}{n}$. We remark, however, two potential inefficiencies in this derivation. First, the upper bound $1 - \frac{1}{n}$ simply corresponds to $AX_k \approx I$ (see Subsection VI-A) thus, we should be able to find ρ_{upper} that converges to a perhaps simpler function of the spectrum of the identity matrix. Second, on the derivation for the rate upper bound in [1] (proof of Thm. 9.1) the sequence of inequalities can be optimized (e.g. we only use the first inequality of such proof) by setting

$$p_i = \frac{\lambda_{max}(S_i^T A S_i)}{\lambda_{max}(\mathcal{S}^T A \mathcal{S})} \qquad (13)$$

yields the following result.

Theorem 2 (A tighter upper bound for adaRBFGS's adaptive rate)**.** *Let ρ_k be the adaptive rate of adaRBFGS as in Thm. 9.1, we can obtain the following upper bound by setting a convenient sampling*

of the form $p_i = \frac{\lambda_{max}(S_i^T A S_i)}{\lambda_{max}(\mathcal{S}^T A \mathcal{S})}$*:*

$$\begin{aligned} \rho_{upper} &= 1 - \kappa(AX_k)^{-1} \\ &= 1 - \frac{\lambda_{min}(AX_k)}{\lambda_{max}(AX_k)} \\ &\geq \rho_k \end{aligned} \tag{14}$$

Proof.

$$\begin{aligned} \rho_k &\geq \lambda_{\min}(A^{1/2} L_k L_k^T A^{1/2}) \lambda_{\min}(D^2) \\ &= \frac{\lambda_{\min(AX_k)}}{\lambda_{\max}(D^{-2})} \\ &= \frac{\lambda_{\min(AX_k)}}{\max_i \lambda_{\max}(S_i^T A S_i)/p_i} \\ &= \frac{\lambda_{\min(AX_k)}}{\lambda_{\max}(AX_k)} = \kappa(AX_k)^{-1} \end{aligned} \tag{15}$$

The first inequality follows from the factorization for $\mathbf{E}\big[Z|X_k\big]$ as detailed in Proposition 7.1 in [1]. Then, in the first equality we use the fact that AX_k and $A^{1/2}X_kA^{1/2}$ have the same spectrum. The second equality is simply because the maximum eigenvalue of a block diagonal matrix is the maximum among maximum eigenvalues of its block matrices. The third equality follows from our choice of sampling probabilities and the fact that $\mathcal{S} = L_k$ (as explained in Subsection IV-C). □

First note that the sampling we propose in (13) is in principle more expensive to calculate to the one in (12). However, (13) is easier to analyze and we can thus get the results from Section VI. In addition, we have not noticed much difference with respect to the computational burden for the experiments we performed in Section VI by adopting (13) instead of (12). Now, one can then outline the advantage of p_i having the form in (13) by the experiments in Figures 8 and 7. First, note in those experiments that ρ_{upper} for Gower's p_i yields loose upper bounds. Hence a lower bound is useful to realize the potential value of the true rate as in Subsection VI-C. On the other hand, observe (Fig. 8) that our proposed ρ_{upper} is much tighter (at all times) with respect to the true rate. In addition, note that the sampling we propose can actually make the true rate hit much lower values (in Fig.8 this corresponds to $1-\rho$ hitting 1) which is not the case for the form of p_i in equation (12) (Fig. 7)

Moreover, the theoretical analysis is simpler and we can state the following result as a corollary from Thm. 2.3

$$\frac{\lambda_{min}(AX_k)}{\lambda_{max}(AX_k)} \leq \frac{\lambda_{min}(AX_{k+1})}{\lambda_{max}(AX_{k+1})} \tag{16}$$

This means that our proposed rate upper bound is non-increasing. The result in (16) is simply because from Thm.2.3 one can see that the largest eigenvalue of AX_k can only decrease while the smallest eigenvalue can only increase. Using (13), we see from Thm. 2 that our upper bound can only decrease at each iteration. Note that such concise conclusion is rather difficult if we work with (12). Indeed, in such scenario the interlacing theorem does not yield an immediate result with respect to the monotonicity for the rate upper bound. This is because the first upper bound (10) (obtained in [1]) depends on the trace of AX_k on the denominator. Then, we do not have a guarantee on the monotonicity of such trace (at least not from Thm. 2.3 because the spectrum of AX_k has eigenvalues both below and above 1) and we cannot conclude something about the monotonicity of the bound in (10).

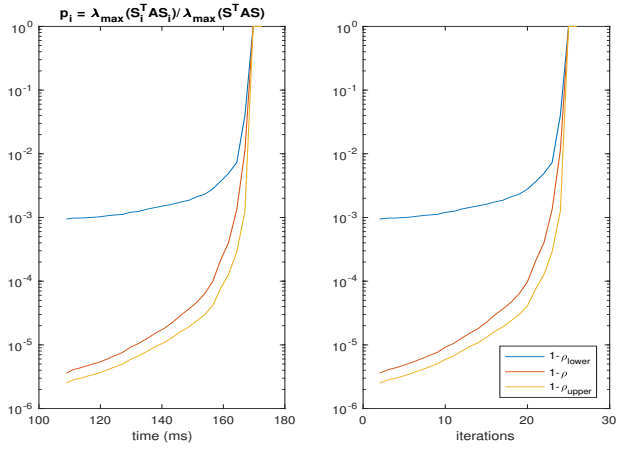

Fig. 8: $1-\rho_k = \lambda_{\min}(A^{-1/2}\mathbf{E}\big[Z|X_k\big]A^{-1/2})$ and lower/upper bounds (as in equations (11) and (14) respectively). These plots correspond to the column variant of adaRBFGS with p_i as in (13).

A non-increasing rate upper bound (as the one we propose) has an immediate consequences (as pointed out in [15]). First, the adaptive version of

RBFGS (e.g. adaRBFGS) can only be better (in expectation) than the non-adaptive version.

Corollary 1 (AdaRBFGS can only be better, in expectation, than RBFGS)**.** *If* $\lambda_{min}(A) \leq 1$ *then* $\prod_{i=0}^{k-1} \rho\prime_i \leq \rho\prime_0^k$.
Where at iterate k *we have* $\rho_k\prime = \rho_{upper} = 1 - \kappa(AX_k)^{-1}$

Proof. Set $\rho\prime_0$ as the rate of the non-adaptive RBFGS and the result follows immediately from inequality (16) above. □

D. Convergence

As we saw in our experiments (and conjectured in [1]) the eigenvalues might converge to a distribution centered at 1 and with very low variance.
We have found, in fact, that a good framework for studying such behavior is given by some well-known results from Random Matrix theory (which we borrow from [10]).
In order to explain how can adaRBFGS make progress at each iteration we refer to the *eigenvalue repulsion* property (Exercise 1.3.15 in [10]). This is, actually, a direct consequence of the interlacing spectrum theorem. In addition and it reveals a fundamental relation between eigenvalues and how these change as we iterate. The property we present (eigenvalue repulsion) is in fact used to prove some of the main results in [10] (see for example the Dyson Brownian motion characterization of the spectrum). We can easily see such property in the case of a rank-1 update.

Corollary 2 (Eigenvalue repulsion, Tao 2011)**.** *Let* λ *be an eigenvalue for* AX_k *that is different from all eigenvalues for* AX_{k+1}*, and w.l.o.g assume* AX_{k+1} *is a minor of* AX_k *obtained by removing the right-most column and last row of* AX_k*. We can then write:*

$$(AX_k)_{n,n} - \lambda = \sum_{j=1}^{n-1} \frac{(u_j(AX_{k+1})^T R)^2}{\lambda_j(AX_{k+1}) - \lambda} \tag{17}$$

Where u_j *is the* j^{th} *eigenvector and* X *is a vector we cut from* AX_k *to get to* AX_{k+1} *as its minor (equivalent definition for compression in Section 4.3 of [7]).*

Proof. We have $AX_k u = \lambda u$ where $u^T = [u_t^T, u_b]$ (u being λ's eigenvector), $u_t \in \mathbb{R}^{n-1\times 1}$ and $u_b \in \mathbb{R}$. By expanding the matrix multiplications in blocks we have $AX_{k+1}u_t + Ru_b = \lambda u_t \Rightarrow$

$$(AX_{k+1} - \lambda I)u_t = -u_b R \tag{18}$$

Because we maintain symmetric and positive definite iterates X_k (c.f. (5)) we know $A^{1/2}X_{k+1}A^{1/2}$ is symmetric and with positive eigenvalues. Hence, we can write $(AX_{k+1} - \lambda I)^{-1} = \sum_j^{n-1} \frac{1}{\lambda_j(AX_{k+1}-\lambda}u_j(AX_{k+1})u_j(AX_{k+1})^T$ (e.g. its spectral decomposition).
Then, from the last equality in the system $AX_k u = \lambda u$ we get $u_b((AX_k)_{n,n} - \lambda) = -R^T u_t$. From (18) we have $(AX_k)_{n,n} - \lambda = R^T(AX_{k+1} - \lambda I)^{-1}R$ and the result follows. □

Note that the result above holds without lost of generality because in the case when AX_{k+1} is a minor that is obtained by removing multiple columns from AX_k then R is just a column block matrix and we just have a larger system of equations. See from the above formula that λ is a rational function with removable poles at $\lambda_j(AX_{k+1})$. This means that as long as the update direction from AX_k to AX_{k+1} does not lead to an orthogonal eigenvector to R then eigenvalues repel each other between iterations.
It turns out that there is a set of results in Random Matrix theory that characterize the distribution of eigenvalues and these stem from an interlacing spectrum. For this purpose, we work with the Empirical Spectrum Distribution (ESD). Namely,

Definition 1 (Empirical spectral distribution)**.**

$$\mu_{AX_k} = \frac{1}{M_k}\sum_{j=1}^{M_k} \delta_{\frac{1}{\sqrt{M_k}}\lambda_j(AX_k)} \tag{19}$$

Where M_k *be the multiplicity of the orthogonal eigenspace (e.g.* n *minus multiplicity of* 1 *as eigenvalue of* AX_k*) to the one corresponding to* 1 *for* AX_k*.*

We scale the eigenvalues in (19) by the inverse square root to properly scale the variance as it is typical in estimators that consist of sums (details in [10]). Now, the reason why we use M_k is that

matrices resulting from compression (e.g. equation (8) which yields Thm.2.3) can be alternatively characterized as minors of the matrix that is compressed (relation show in Section 4.3 of [7]). Such characterization is interesting because of the results that we can adapt from [10] to our setting (see Section VI). From now on, we can think of the sequence AX_k as successive minors where the number of columns we eliminate to go from one to the other (and rows accordingly) is $M_k - M_{k+1} \leq q$ (where such bound is obtained in Proposition 2.8 from [6]). The main results that concern us from [10] are those that involve the stability of the ESD and its asymptotic convergence properties.

A first result is the *Stability of ESD laws with respect to small rank perturbation* for symmetric matrices (note that the objective in (4) acts as a regularizer, thus ensuring that the difference between iterates is a low rank update). What this result says when applied to our setting is the following.

Conjecture 1 (Strong convergence of the ESD for the spectrum of AdaRBFGS iterates)**.** $\mu_{AX_{k+1}} \xrightarrow{a.s.} \mu$ *for fixed* μ*.*

We make this conjecture because of stronger results related to 3 appear in [10] as direct consequences of an interlacing spectrum (e.g. Thm. 2.3). For example, it can be shown that to establish the circular law (analogous of Central Limit theorem for random matrices) it suffices to use a similar result to 3 (c.f. exercise 2.4.2 in [10]). Hence, it is plausible that the ESD for AX_k converges to some fixed distribution.

However, we do not have a explicit expression for μ in our setting because at the moment this theory is beyond the scope of this project. As an illustration we mention a generalization of the above result which is the Semicircular law (for sequences of Wigner matrices) which gives an explicit formula for μ. While Wigner matrices are more of a theoretical construction (infinite matrices), there are some results in [10] that make the connexion with finite i.i.d and non i.i.d matrices.

A second set of results from [10] that are relevant to convergence of adaRBFGS are the (weak-strong) versions of the Bai-Yin theorem (c.f. Theorem 2.3.23 in [10]). What is important, for us, about Bai-Yin theorems (applied to real symmetric matrix sequences) is that these show that the spectral radius (λ_{max} in our setting) of AX_k is asymptotically bounded by a constant (asymptotically almost surely). Then, the convergence of the ESD and Bai-Yin theorems altogether would imply that AX_k is well conditioned asymptotically almost surely.

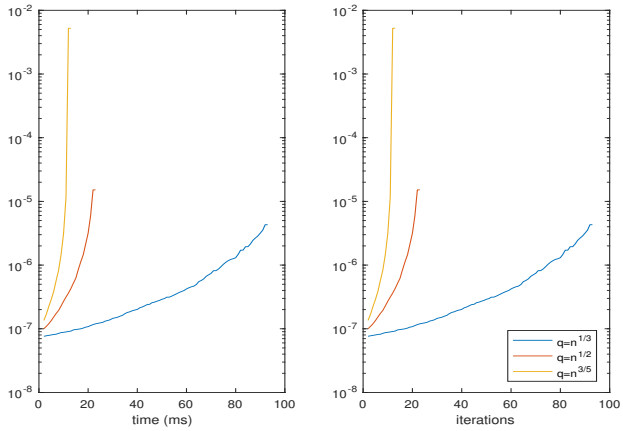

Fig. 9: Rates for adaRBFGS (column variant) as a function of the dimension of the sketch (e.g. q).

E. Dependence on sketch dimension

Our last result concerns a particular behavior we have observed during the experiments which is the dependence of the number of iterations on the sketch dimension q. A first result in this direction is the lower bound presented in [1] for the non-adaptive RBFGS (constant rate), which is

$$1 - \frac{\mathbf{E}[q]}{n} \leq \rho \tag{20}$$

What inequality (20) implies is that increasing the sketch dimension q can only make the constant rate smaller. Nonetheless, there is a clear trade-off with respect to the complexity per iteration (floating point operations) which is dominated by a matrix inversion (see update rule in Section IV) of order q e.g. $O(q^3)$. Now, we want to know how the sketch dimension depends on the rate for the adaptive setting (e.g. adaRBFGS). From numerical experiments as in Fig. 9 we see that the $1 - \rho_k$

can only converge faster as we increase q. One can justify this observation with a result stemming (again) from Thm. 2.3 which is a characterization of the stability of the ESD (which comes from 2.4.1 in [10]).

Corollary 3 (Stability of ESD)**.** *Let M_k and μ_{AX_k} as defined in (19). We then have:*

$$\begin{aligned}\frac{M_k}{M_{k+1}}\mu_{AX_k}([-\infty,\lambda]) - \frac{q}{M_{k+1}} \\ \leq \mu_{AX_{k+1}}([-\infty,\lambda]) \\ \leq \frac{M_k}{M_{k+1}}\mu_{AX_k}([-\infty,\lambda])\end{aligned} \tag{21}$$

with $0 < c < 1$.

Proof. First we prove the second inequality. By Thm. 2.3, we have that for each $i \in [n-q]$ we have $\lambda_i(AX_{k+1}) \leq \lambda_{i+q}(AX_k)$. Then, $\frac{M_{k+1}}{M_k}\frac{1}{\sqrt{M_{k+1}}}\lambda_i(AX_{k+1}) \leq \frac{1}{\sqrt{M_k}}\lambda_{i+q}(AX_k)$ as $M_k \geq M_{k+1}$ by Proposition 2.8 in [6]. And the inequality follows (point-wise) from the definition of ESD in (19).

For the first inequality we proceed as for the second inequality but using $\lambda_i(AX_k) \leq \lambda_i(AX_{k+1})$ for $i \in [n-q]$. Now, note that because $M_k \geq M_{k+1}$ the contribution of those additional eigenvalues (e.g. difference in cardinality between the orthogonal eigenspace to 1 for AX_k and AX_{k+1} respectively) can make $\mu_{AX_k}[-\infty,\lambda] \geq \mu_{AX_{k+1}}[-\infty,\lambda]$. However, we can bound the amount of such additional eigenvalues by Proposition 2.8 of [6] as follows: $M_k - M_{k+1} \leq q$. Then, the contribution of these additional eigenvalues to μ_{AX_k} is bounded by $\frac{q}{M_{k+1}} \geq \frac{q}{M_k}$ and the result follows. □

Observe from this last result that when q is large then the ESD can change quite a lot from one iteration to the other. However, for q very small the ESD is stable (e.g. changes very little) in one iteration. In this case having an unstable ESD (large q) can be beneficial. For example, say we have large $\lambda_{max}(A)$ meaning that we want to eliminate, as quickly as possible, the tail (which implies a large condition number for AX_k) composed of large eigenvalues. In the latter scenario, a stable transition in eigenvalue distribution could take quite long to eliminate a long tail (as we can confirm in Fig. 9) as we make little progress per iteration. A part from Corollary 3, the factorization in (8) offers some complementary intuition on the observed experimental results. That is, we see in (8) that the number of eigenvalues to compress is $n-q$ (e.g. can only be smaller for larger q). These eigenvalues seem to converge, nonetheless, to a distribution centered at 1 (c.f. VI-D) independent of the dimension of the sketch q. Hence, to preserve the variance (same distribution in all cases) of such limiting distribution, less eigenvalues (e.g. large q) should be more spread compared to the situation when we have more eigenvalues to compress (e.g. smaller q). Now, the jumps that eigenvalues can make from iteration k to $k+1$ is bounded by the distances between eigenvalues for AX_k from Thm. 2.3. Hence, having larger distances between eigenvalues in AX_k and an unstable (c.f. 3) transitions can eliminate those unwanted long eigenvalue tails quicker.

Lastly, we mention that the latter remarks do not imply that the number of iterations monotonically decreases with q. This is because we base the argument on bounds for the eigenvalue jumps, and change in distribution. Moreover, the convergence of the ESD to a fixed distribution is still an hypothesis (e.g. Conjecture 1). Hence, a proof of Conjecture 1 is a possible direction for future research as well as the possibility of monotonic behavior in the number of iterations with respect to q.

VII. Conclusion

In this project we have analyzed the rate for adaRBFGS which is a new promising method that can be used as a primitive for first and second order methods. In the process we have surveyed related literature that serves as basis for adaRBFGS and pointed out some additional results that are useful for a finer grained analysis of the algorithm's rate (which were apparently unnoticed by the authors in [1]). A first such result and the starting point of our analysis is the interlacing spectrum theorem (see Subsection VI-A). Such property suggests that the spectrum of AX_k shrinks as we iterate, and that eigenvalues accumulate at 1 (which was conjectured

in [1]). In addition, we propose a different convenient sampling strategy (to sample sketch matrices for adaRBFGS) for which we get (c.f. VI-C) a simpler theoretical analysis for the rate (using the interlacing theorem). We observed as consequence of this proposed sampling strategy that both a lower and upper bound we derived look, experimentally, much tighter than previously stated bounds in [1]. Then, as direct consequence of the form of our upper bound and the interlacing theorem we saw that the rate upper bound is non-increasing and that adaRBFGS outperforms RBFGS (non adaptive version). We point out, however, that we observed experimentally that the true rate ρ_k is also non-increasing. However, the proof of such result is not as straightforward. In fact, if this result is of interest for future result we suggest looking at the relation between subsequent column matrices to be sampled from L_k and L_{k+1}, namely S_i^k and S_i^{k+1}. It is not hard to see that (from the Courant-Fisher characterization of eigenvalues) that if one could prove either $S_i^k \succ S_i^{k+1}$ (or the opposite) at each iteration, then ρ_k is non-increasing. Moreover, ρ_k for adaRBFGS seems to have better convergence behavior in our experiments. With these results in mind, we embarked on the analysis of convergence for the rate. For this purpose we borrowed results from random matrix theory. This is because such theory has a rich description on the asymptotic and stochastic properties for the spectrum of sequence of matrices such as those we generate for adaRBFGS. Namely, these results describe stability and convergence modes for the Empirical Spectrum Distribution (ESD), that gives us additional theoretical justification on the experimental results (centered distributions around 1) we see. While the results we used are rather general, there is potentially room for specialization as the ones existing for other type of matrices (such as the semi-circle law in [10]). Lastly, we used the stability of the ESD to investigate the relation between the sketch dimension q and the true adaRBFGS rate. While there is a result for the non-adaptive version (Theorem 6.1 in [1]) there was no obvious way to extend this to adaRBFGS which is why we investigated this behavior. A next step would be to make explicit the trade-off between the number of iterations and flops per iteration in terms of q and n (number of dimensions). Such question is appealing as it would answer a very practical question for the use of adaRBFGS: how to choose q that is most efficient for my problem?. Lastly, we mention an interesting direction for future research. This consists of going beyond proving that the rate (or its upper bound) are non-increasing (as proved in Subsection VI-C) and finding the sequence of decrement ratios between subsequent rates. Such a result is plausible as the convergence time depends mainly on $\lambda_{max}(AX_k)$ and a set of results for bounding the spectral radius have been derived using related to the interlacing spectrum (c.f. [13], [14]).

REFERENCES

[1] *Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms.* Robert M. Gower and Peter Richtárik. arXiv:1602.01768, 2016.

[2] *Randomized iterative methods for linear systems.* Robert M. Gower and Peter Richtárik. SIAM Journal on Matrix Analysis and Applications 36.4, pp. 1660-1690, 2015.

[3] *Stochastic Block BFGS: Squeezing More Curvature our of Data.* R.M. Gower, D. Goldfarb and P. Richtárik. Proceedings of The 33rd International Conference on Machine Learning, PMLR 48:1869-1878, 2016.

[4] *Linearly convergent randomized iterative methods for computing the pseudoinverse.* Robert M. Gower and Peter Richtarik. Preprint, December 2016.

[5] *A suite of randomized methods for inverting positive definite matrices implemented in MATLAB.* Robert M Gower. http://www.maths.ed.ac.uk/~prichtar/i_software.html.

[6] *On a class of limited memory preconditioners for large-scale nonlinear least-squares problems.* SIAM Journal on Optimization 21.3, pp. 912-935, 2011.

[7] *Matrix Analysis.* R.A. Horn and C.R. Johnson. Cambridge University Press, Cambridge, England, 1999.

[8] *A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms.* L. Nazareth. SIAM Journal on Numerical Analysis, 16:794-800, 1979.

[9] *A new approach to variable metric algorithms.* R. Fletcher. Computer Journal, 13:317-322, 1970.

[10] *Topics in Random Matrix Theory.* Terrence Tao. Graduate Studies in Mathematics, Volume 132. American Mathematical Society, 2012.

[11] *Eigenvalue Inequalities for Matrix Product.* Fuzhen Zhang and Qingling Zhang. IEEE Transactions on Automatic Control, 2006.

[12] *The Cauchy interlacing theorem in simple Euclidean Jordan algebras and some consequences.* M. Seetharama Gowda and J. Tao. Linear and Multilinear Algebra, 16:41, 2009.

[13] *Cauchy's interlace theorem and lower bounds for the spectral radius.* A.Mcd. Mercer and Peter R. Mercer. International Journal of mathematics and Mathematical Sciences. Vol. 23, Issue 8, pp. 563-566, 2000.

[14] *Lower bounds for the spectral radius of a matrix.* Bill G. Horne. NECI Technical Report 95-14, NEC Research Institute, 1995.

[15] Sebastian U. Stich, Robert M. Gower and Peter Richtárik. *Complexity of Adaptive Randomized BFGS for Matrix Inversion.* Preprint, 2016.

[16] *Variable metric random pursuit.* S. U. Stich, C.L. Muller, and B.Gartner. Mathematical Programming 156.1 (2015), pp. 549-579.

[17] *Randomized Hessian estimation and directional search.* D.Leventhal and A.Lewis. Optimization 60.3(2011), pp. 329-345.

[18] *Accelerating stochastic gradient descent using predictive variance reduction.*Rie Johnson, Tong Zhang. Advances in Neural Information Processing Systems. pp. 315-323, 2013.