RADAGRAD: Random Projections for Adaptive Stochastic Optimization

Gabriel Krummenacher* **Brian McWilliams***
Department of Computer Science
ETH Zürich, Switzerland
{gabriel.krummenacher|mcbrian}@inf.ethz.ch

Abstract

We present RADAGRAD a simple and computationally efficient approximation to full-matrix ADAGRAD based on dimensionality reduction using the subsampled randomized Hadamard transform. RADAGRAD is able to capture correlations in the gradients and achieves a similar regret – in theory and empirically – to full-matrix ADAGRAD but at a computational cost comparable to the diagonal variant.

1 Introduction

ADAGRAD [5] is a stochastic sub-gradient optimization method which adaptively sets the learning rate for each feature by means of a time-varying *proximal* regularizer. It has gained extensive popularity in particular for the large-scale optimization problems inherent in learning the parameters for deep neural networks.

The full matrix variant of ADAGRAD (which we refer to as ADAGRAD-F) uses the root of the outer products of the gradients as the proximal term. For computational reasons, the most commonly studied and utilised version of ADAGRAD considers only a diagonal matrix proximal term (which we refer to as ADAGRAD-D).

In this work we propose RADAGRAD: A randomized approximation to ADAGRAD-F where the full proximal matrix is approximated using a fast Johnson-Lindenstrauss projection, specifically the Subsampled Randomized Hadamard Transform (SRHT). We show that RADAGRAD achieves performance close to full ADAGRAD at a fraction of the computational cost.

Problem setting. The problem we tackle is online stochastic optimization where the goal is, at each step, to predict a point $\boldsymbol{\beta}_t \in \mathbb{R}^p$ which achieves low regret with respect to a fixed optimal predictor, $\boldsymbol{\beta}^{\text{opt}}$, for a sequence of (convex) functions $F_t(\boldsymbol{\beta})$. After T rounds, the regret can be defined in the following way

$$R(T) = \sum_{t=1}^{T} F_t(\boldsymbol{\beta}_t) - \inf_{\boldsymbol{\beta}} \sum_{t=1}^{T} F_t(\boldsymbol{\beta}) = \sum_{t=1}^{T} F_t(\boldsymbol{\beta}_t) - \sum_{t=1}^{T} F_t(\boldsymbol{\beta}^{\text{opt}}). \tag{1}$$

We will consider functions F_t of the form $F_t(\boldsymbol{\beta}) := f_t(\boldsymbol{\beta}) + \varphi(\boldsymbol{\beta})$ where f_t and φ are convex loss and regularization functions respectively. Throughout, the vector $\mathbf{g}_t \in \partial f_t(\boldsymbol{\beta}_t)$ refers to a particular subgradient of the loss function. Standard first order methods update $\boldsymbol{\beta}_t$ at each step by moving in the opposite direction of $\mathbf{g}_t$ according to a step-size parameter, η. The ADAGRAD family[1] of algorithms [5] instead use an *adaptive* learning rate which can be different for each feature. This is controlled using a time-varying proximal term which we briefly review. Defining $\mathbf{G}_t = \sum_{i=1}^{t} \mathbf{g}_i \mathbf{g}_i^\top$ and

$$\mathbf{H}_t = \delta \mathbf{I}_p + (\mathbf{G}_{t-1} + \mathbf{g}_t \mathbf{g}_t^\top)^{1/2},$$

*Authors contributed equally.

[1]For brevity we only consider ADAGRAD for regularized dual averaging here.

the ADAGRAD-F proximal term is given by $\psi_t(\boldsymbol{\beta}) = \frac{1}{2}\langle \boldsymbol{\beta}, \mathbf{H}_t\boldsymbol{\beta}\rangle$. The update at time $t+1$ is given by the following optimization problem

$$\boldsymbol{\beta}_{t+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \eta \langle \bar{\mathbf{g}}_t, \boldsymbol{\beta}\rangle + \eta\varphi(\boldsymbol{\beta}) + \frac{1}{t}\psi_t(\boldsymbol{\beta}) \right\}, \tag{2}$$

with $\bar{\mathbf{g}}_t = \frac{1}{t}\sum_{i=1}^{t}\mathbf{g}_i$, the average gradient vector at time t.

Clearly when p is large, constructing $\mathbf{G}$ and finding its root and inverse *at each iteration* is impractical. In practice, rather than the full outer product matrix, ADAGRAD-D uses a proximal function consisting of the diagonal of $\mathbf{G}_t$, $\psi_t(\boldsymbol{\beta}) = \frac{1}{2}\left\langle \boldsymbol{\beta}, \left(\delta\mathbf{I}_p + \operatorname{diag}(\mathbf{G}_t)^{1/2}\right)\boldsymbol{\beta}\right\rangle$.

Whilst the diagonal proximal term is computationally cheaper, it is unable to capture dependencies between coordinates in the gradient terms – a typical assumption is that these terms are independent. As an interpolation between the computational simplicity of the diagonal proximal term and the rich representation afforded by the full-matrix term, [5] proposed block diagonal proximal terms, although this effectively still assumes independence between blocks of coordinates. Aside from its good empirical performance, ADAGRAD-D has been shown to achieve optimal regret in a minimax sense under certain assumptions about data sparsity [6].

In the following we review the fast Johnson-Lindenstrauss projection which will form the basis for our randomized approximation scheme.

Fast Johnson-Lindenstrauss projections. Johnson-Lindenstrauss (J-L) projections are low-dimensional embeddings $\boldsymbol{\Pi} : \mathbb{R}^p \to \mathbb{R}^\tau$ which preserve – up to a small distortion – pairwise ℓ_2 distances between vectors according to the J-L lemma (see e.g. [2]). Typically, $\boldsymbol{\Pi} \in \mathbb{R}^{\tau\times p}$ is constructed to have nearly-orthogonal rows with entries drawn at random from a sub-gaussian distribution [1]. We concentrate on the class of *structured* random projections, among which the Subsampled Randomized Hadamard Transform (SRHT) has received particular recent attention [3, 10]. The SRHT consists of a preconditioning step after which τ columns of the new matrix are subsampled uniformly at random as $\boldsymbol{\Pi} = \sqrt{p/\tau}\mathbf{S}\boldsymbol{\Theta}\mathbf{D}$ [3] with the definitions:

- $\mathbf{S} \in \mathbb{R}^{\tau\times p}$ is a subsampling matrix.
- $\mathbf{D} \in \mathbb{R}^{p\times p}$ is a diagonal matrix whose entries are drawn independently from $\{-1, 1\}$.
- $\boldsymbol{\Theta} \in \mathbb{R}^{p\times p}$ is a normalized Walsh-Hadamard matrix[2] which is defined recursively as

$$\boldsymbol{\Theta}_p = \begin{bmatrix} \boldsymbol{\Theta}_{p/2} & \boldsymbol{\Theta}_{p/2} \\ \boldsymbol{\Theta}_{p/2} & -\boldsymbol{\Theta}_{p/2} \end{bmatrix}, \quad \boldsymbol{\Theta}_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}.$$

  We set $\boldsymbol{\Theta} = \frac{1}{\sqrt{p}}\boldsymbol{\Theta}_p$ so it has orthonormal columns.

Analogously to the fast Fourier transform (FFT), applying the SRHT to a $p-$dimensional vector can be achieved in $O(p\log\tau)$ time. Recently the SRHT has gained popularity as a way to speed up [8] and robustify [9] large-scale linear regression.

2 A randomized approximation to $\mathbf{H}_t$

We propose RADAGRAD: A variant of ADAGRAD-F which uses an approximation to the outer product matrix based on the SRHT. Defining $\tilde{\mathbf{G}}_t = \sum_{i=1}^{t}(\boldsymbol{\Pi}\mathbf{g}_i)(\boldsymbol{\Pi}\mathbf{g}_i)^\top$, we consider the randomized approximation to $\mathbf{H}_t$,

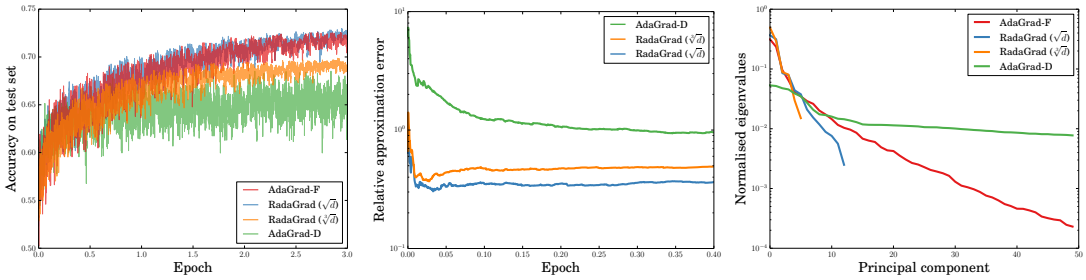
$$\tilde{\mathbf{H}}_t \in \mathbb{R}^{\tau\times\tau} = \delta\mathbf{I}_\tau + (\tilde{\mathbf{G}}_{t-1} + (\boldsymbol{\Pi}\mathbf{g}_t)(\boldsymbol{\Pi}\mathbf{g}_t)^\top)^{1/2},$$

where $\boldsymbol{\Pi} \in \mathbb{R}^{\tau\times p}$ is a random projection which reduces the dimensionality of $\mathbf{g}_t$ to $\tau \ll p$ so that constructing $\tilde{\mathbf{G}}_t$ and computing its root and inverse is inexpensive.

Analogously to ADAGRAD-F, the RADAGRAD proximal term then is $\tilde{\psi}_t(\boldsymbol{\beta}) = \frac{1}{2}\left\langle \boldsymbol{\Pi}\boldsymbol{\beta}, \tilde{\mathbf{H}}_t\boldsymbol{\Pi}\boldsymbol{\beta}\right\rangle$. The RADAGRAD update at time $t+1$ is given by the following optimization problem

$$\boldsymbol{\beta}_{t+1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \eta \langle \bar{\mathbf{g}}_t, \boldsymbol{\beta}\rangle + \eta\varphi(\boldsymbol{\beta}) + \frac{1}{t}\tilde{\psi}_t(\boldsymbol{\beta}) \right\}, \tag{3}$$

[2]For the Hadamard transform, p must be a power of two but other transforms exist (e.g. DCT, DFT) with similar theoretical guarantees and no restriction on p.



(a) (b) (c)

Figure 1: Comparison of: (a) test error, (b) relative approximation error of the trace of the proximal term and (c) the largest eigenvalues (normalised by their sum) of the proximal term.

Solving (3) will typically involve taking the inverse of $\mathbf{\Pi}^\top \tilde{\mathbf{H}}_t \mathbf{\Pi}$. Due to the structure of $\mathbf{\Pi}$, this reduces to simply inverting the $\tau \times \tau$ matrix $\tilde{\mathbf{H}}_t$ and applying an "inverse" SRHT transformation (analogous to the inverse FFT). Due to the J-L property, $\tilde{\mathbf{H}}_t$ is still able to retain information about correlations between *all* coordinates in the gradient vectors. Importantly, we still obtain a solution in the original space, $\mathbb{R}^p$ since RADAGRAD still obtains the original gradient vectors $\mathbf{g}_t$ at each step and the random projection is only necessary to approximate the proximal term. We will clarify this in the following section with a simple example.

2.1 RADAGRAD with squared-ℓ_2 regularisation

As an illustrative example, we derive the explicit update equation for the simple but useful case of squared ℓ_2 regularization. For $\varphi(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|_2^2$, the optimisation problem in Equation (3) can be solved analytically:

$$\boldsymbol{\beta}_{t+1} = -\left(2\lambda \mathbf{I}_p + \frac{1}{t\eta} \mathbf{\Pi}^\top \tilde{\mathbf{H}}_t \mathbf{\Pi}\right)^{-1} \bar{\mathbf{g}}_t. \tag{4}$$

Using the Woodbury matrix identity, the inverse in Equation (4) can be computed efficiently:

$$\boldsymbol{\beta}_{t+1} = -\frac{1}{2\lambda}\left(\bar{\mathbf{g}}_t - \frac{1}{t\eta}\frac{1}{2\lambda}\mathbf{\Pi}^\top \left(\tilde{\mathbf{H}}_t^{-1} + \frac{p}{2\tau\lambda t\eta}\mathbf{I}_\tau\right)^{-1} (\mathbf{\Pi}\bar{\mathbf{g}}_t)\right) \tag{5}$$

since $\mathbf{\Pi}\frac{1}{2\lambda}\mathbf{\Pi}^\top = \frac{p}{2\tau\lambda}\mathbf{I}_\tau$. Eq. (5) requires the random projection of the vector of average gradients, $\mathbf{\Pi}\bar{\mathbf{g}}_t$. If we denote $\tilde{\bar{\mathbf{g}}}_{t-1} = \mathbf{\Pi}\bar{\mathbf{g}}_{t-1}$, this can be computed easily as $\mathbf{\Pi}\bar{\mathbf{g}}_t = \tilde{\mathbf{g}}_{t-1} + \frac{\mathbf{\Pi}\mathbf{g}_t - \tilde{\mathbf{g}}_{t-1}}{t}$. Therefore, at each step we only require a single random projection of the gradient vector and a single up-projection of a τ-dimensional vector. This procedure is summarized in Algorithm 1.

Clearly, some more care is required to derive efficient update equations for different regularization functions which we will make explicit in the full version of this work.

Computational complexity. Each SRHT costs $O(p \log p)$. Constructing $\tilde{\mathbf{G}}_t$ costs $O(\tau^2)$. The main cost at each iteration is computing the root of $\tilde{\mathbf{G}}_t$ and the inverse of $\tilde{\mathbf{H}}_t$. Each of these operations costs $O(\tau^3)$. In comparison, the per-iteration cost of ADAGRAD-F is $O(p^3)$ whereas ADAGRAD-D costs $O(p)$ per iteration. Therefore, with careful choice of τ we arrive at an algorithm with minimal overhead compared with ADAGRAD-D.

2.2 Regret bound of RADAGRAD

Unsurprisingly, by application of the J-L property of the SRHT [10], RADAGRAD can be shown to entertain a regret bound similar to ADAGRAD-F. We sketch this in the following theorem.

Theorem 1. *Under the definition in Eq.* (1)*,* RADAGRAD *achieves the following regret with high probability*

$$R(T) \leq \frac{\delta}{\eta}(1+\rho)\|\boldsymbol{\beta}^{opt}\|_2^2 + \frac{1}{\eta}(1+\rho)^{3/2}\|\boldsymbol{\beta}^{opt}\|_2^2 \cdot \mathrm{tr}(\mathbf{G}_T^{1/2}) + \eta\sqrt{(1+\rho)} \cdot \mathrm{tr}(\mathbf{G}_T^{1/2}).$$

The distortion factor ρ depends on the projection dimension, the rank of $\tilde{\mathbf{G}}_T$ and the desired probability of error (according to Lemma 3.4 of [10]).

3 Identifying defective train wheels

The task is to classify wheels of freight trains on the Swiss rail network as defective or non-defective. The data collected is similar to [7] except the problem is treated as fully supervised. The data comprises of vertical force measurements of $n = 2,293$ wheels. For each measurement we extracted $p = 144$ features by computing empirical moments of coefficients in a 10-level multi-scale wavelet transform.

We choose $f_t(\boldsymbol{\beta})$ to be the hinge loss. As in the example in §2.1, $\varphi_t(\boldsymbol{\beta})$ is the squared ℓ_2 regularizer. The problem is low-dimensional so results for ADAGRAD-F can be obtained. The regularization strength, λ and the step size η are identical for all three methods. We show results for two different projection dimensions. We choose $\tau = p^{\frac{1}{2}}$ so construction and storage of $\tilde{\mathbf{H}}_t$ is equivalent in cost to ADAGRAD-D, and $\tau = p^{\frac{1}{3}}$ so the cost of computing $\tilde{\mathbf{G}}_t^{1/2}$ and $\tilde{\mathbf{H}}_t^{-1}$ is equivalent to ADAGRAD-D.

Algorithm 1 RADAGRAD

Input: $\eta > 0, \delta \geq 0, \tau$

for $t = 1 \dots T$ **do**
  Receive $\mathbf{g}_t \in \partial f_t(\boldsymbol{\beta}_t)$.
  Project: $\tilde{\mathbf{g}}_t = \boldsymbol{\Pi}\mathbf{g}_t$
  $\tilde{\mathbf{G}}_t = \tilde{\mathbf{G}}_{t-1} + \tilde{\mathbf{g}}_t\tilde{\mathbf{g}}_t^\top$, $\tilde{\mathbf{S}}_t = \tilde{\mathbf{G}}_t^{\frac{1}{2}}$
  $\bar{\tilde{\mathbf{g}}}_t = \frac{t-1}{t}\bar{\tilde{\mathbf{g}}}_{t-1} + \frac{1}{t}\tilde{\mathbf{g}}_t$
  $\tilde{\mathbf{H}}_t^{-1} = (\delta\mathbf{I}_p + \tilde{\mathbf{S}}_t)^{-1}$
  $\tilde{\mathbf{j}}_t = (\tilde{\mathbf{H}}^{-1} + \frac{1}{t\eta}\frac{p}{2\tau\lambda}\mathbf{I}_\tau)^{-1}\bar{\tilde{\mathbf{g}}}_t$
  Up-project: $\mathbf{j}_t = \boldsymbol{\Pi}^\top\tilde{\mathbf{j}}_t$
  $\boldsymbol{\beta}_{t+1} = -\frac{1}{2\lambda}(\bar{\mathbf{g}} - \frac{1}{t\eta}\frac{1}{2\lambda}\mathbf{j}_t)$
end for

Output: $\boldsymbol{\beta}_T$

Fig. 1a compares the mean prediction accuracy per iteration over 50 trials. ADAGRAD-F achieves better accuracy than ADAGRAD-D as well as reduced variance. The accuracy of RADAGRAD quickly approaches ADAGRAD-F. As expected, decreasing τ, decreases the accuracy of RADAGRAD. Interestingly, RADAGRAD outperforms ADAGRAD-D despite using as few as 6 dimensions (compared with 144) to represent the initial gradient. The reason for this is illuminated in Fig. 1b. The relative difference between $\mathrm{tr}(\mathbf{H}_t)$ and $\mathrm{tr}(\tilde{\mathbf{H}}_t)$ is smaller than the error from using only a diagonal matrix. Since $\mathrm{tr}(\mathbf{H}_t)$ controls the regret of ADAGRAD (Theorem 7 of [5]), the improvement of RADAGRAD over ADAGRAD-D can be explained by the better approximation to this term.

Fig. 1c shows the largest eigenvalues (normalized by their sum) of the proximal matrix for each method computed after one epoch of training. The spectrum of $\mathbf{H}_t$ decays rapidly which is matched by the randomized approximation. This illustrates the dependencies between the coordinates in the gradients and suggests $\mathbf{H}_t$ can be well approximated by a low-dimensional matrix which considers these dependencies. On the other hand the spectrum of ADAGRAD-D decays much slower which explains its larger trace relative to $\mathrm{tr}(\mathbf{H}_t)$.

Currently, the small size of the problem and our un-optimized Python implementation causes the runtime of RADAGRAD to be dominated by constant overhead. Therefore, although both configurations of RADAGRAD are significantly faster than ADAGRAD-F, ADAGRAD-D is still faster per iteration. In practice, RADAGRAD should scale better in high-dimensional problems with many correlated features.

4 Discussion and further work

We have presented RADAGRAD which approximates the full proximal term of ADAGRAD using fast, structured random projections. RADAGRAD achieves similar performance to ADAGRAD-F at a fraction of the computational cost with a marked improvement in prediction accuracy and reduction in variance over diagonal ADAGRAD-D. This is an important result since until now the computational expense of ADAGRAD-F has rendered it impractical for widespread use. ADAGRAD-D, on the other hand, has been widely adopted in many real-world applications – notably the large scale optimization tasks inherent in training deep neural networks (see e.g. [4]). These preliminary results suggest that RADAGRAD is able to bridge the gap between the computational benefit of ADAGRAD-D and the improved theoretical properties of ADAGRAD-F. In our experiments the projection dimension was chosen to draw a direct comparison to the computational cost of ADAGRAD-D. We intend to explore the effect of varying τ on the performance of RADAGRAD.

Recently other adaptive learning rate methods for stochastic optimization have been proposed, for example ADADELTA [11], which often displays better empirical performance than ADAGRAD-D. We aim to perform a full empirical study which compares RADAGRAD with ADADELTA as well as related limited memory approximations to second-order methods such as L-BFGS.

Acknowledgements. We are grateful to David Balduzzi & Christina Heinze for valuable discussions and suggestions and Stefan Koller at SBB for help with the data.

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 2003.

[2] Nir Ailon and Bernard Chazelle. The fast johnson-lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

[3] Christos Boutsidis and Alex Gittens. Improved matrix algorithms via the Subsampled Randomized Hadamard Transform. 2012. arXiv:1204.0062v4 [cs.DS].

[4] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[5] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[6] John C. Duchi, Michael I. Jordan, and H. Brendan McMahan. Estimation, optimization, and parallelism when data is sparse. In *Advances in Neural Information Processing Systems*, 2013.

[7] Gabriel Krummenacher, Cheng S Ong, and Joachim Buhmann. Ellipsoidal multiple instance learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 73–81, 2013.

[8] Michael W Mahoney. Randomized algorithms for matrices and data. April 2011. arXiv:1104.5557v3 [cs.DS].

[9] Brian McWilliams, Gabriel Krummenacher, Mario Lucic, and Joachim M Buhmann. Fast and robust least squares estimation in corrupted linear models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27.

[10] Joel A Tropp. Improved analysis of the subsampled randomized Hadamard transform. November 2010. arXiv:1011.1595v4 [math.NA].

[11] Matthew D Zeiler. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.