Convergence rates of sub-sampled Newton methods

Murat A. Erdogdu* Andrea Montanari*,†

Abstract

We consider the problem of minimizing a sum of n functions via projected iterations onto a convex parameter set $\mathcal{C} \subset \mathbb{R}^p$ where $n \gg p \gg 1$. In this regime, algorithms which utilize sub-sampling techniques are known to be effective. In this paper, we use sub-sampling techniques together with eigenvalue thresholding to design a new randomized batch algorithm which possesses comparable convergence rate to Newton's method, yet has much smaller per-iteration cost. The proposed algorithm is robust in terms of starting point and step size, and enjoys a composite convergence rate, namely, quadratic convergence at start and linear convergence when the iterate is close to the minimizer. We develop its theoretical analysis which also allows us to select near-optimal algorithm parameters. Our theoretical results can be used to obtain convergence rates of previously proposed sub-sampling based algorithms as well. We demonstrate how our results apply to well-known machine learning problems. Lastly, we evaluate the performance of our algorithm on several datasets under various scenarios.

1 Introduction

We consider the problem of minimizing an average of n functions $f_i : \mathbb{R}^p \to \mathbb{R}$,

$$\underset{\theta}{\text{minimize}} \; f(\theta) \coloneqq \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \,, \tag{1.1}$$

in a batch setting, where n is assumed to be much larger than p. Most machine learning models can be expressed as above, where each function f_i corresponds to an observation. Examples include logistic regression, support vector machines, neural networks and graphical models.

Many optimization algorithms have been developed to solve the above minimization problem using iterative methods [Bis95, BV04, Nes04]. In this paper, we consider the iterations of the following form

$$\theta^{t+1} = \theta^t - \eta_t \mathbf{Q}^t \nabla_\theta f(\theta^t), \tag{1.2}$$

where η_t is the step size and $\mathbf{Q}^t$ is a suitable scaling matrix that provides curvature information (For simplicity, we drop the projection throughout the introduction, i.e., we assume $\mathcal{C} = \mathbb{R}^p$).

*Department of Statistics, Stanford University

†Department of Electrical Engineering, Stanford University

Updates of the form Eq. (1.2) have been extensively studied in the optimization literature. The case where $\mathbf{Q}^t$ is equal to the identity matrix corresponds to *Gradient Descent* (GD) which, under smoothness assumptions, achieves linear convergence rate with $\mathcal{O}(np)$ per-iteration cost. More precisely, GD with ideal step size yields

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_{1,\mathrm{GD}}^t \|\hat{\theta}^t - \theta_*\|_2 \,,$$

where, as $\lim_{t\to\infty} \xi_{1,\mathrm{GD}}^t = 1 - (\lambda_p^*/\lambda_1^*)$, and λ_i^* is the i-th largest eigenvalue of the Hessian of $f(\theta)$ at minimizer θ_*.

Second order methods such as *Newton's Method* (NM) and *Natural Gradient Descent* (NGD) [Ama98] can be recovered by taking $\mathbf{Q}^t$ to be the inverse Hessian and the Fisher information evaluated at the current iterate, respectively. Such methods may achieve quadratic convergence rates with $\mathcal{O}(np^2 + p^3)$ per-iteration cost [Bis95, Nes04]. In particular, for t large enough, Newton's Method yields

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_{2,\mathrm{NM}}^t \|\hat{\theta}^t - \theta_*\|_2^2,$$

and it is insensitive to the condition number of the Hessian. However, when the number of samples grows large, computation of $\mathbf{Q}^t$ becomes extremely expensive.

A popular line of research tries to construct the matrix $\mathbf{Q}^t$ in a way that the update is computationally feasible, yet still provides sufficient second order information. Such attempts resulted in Quasi-Newton methods, in which only gradients and iterates are used in the construction of matrix $\mathbf{Q}^t$, resulting in an efficient update at each step t. A celebrated Quasi-Newton method is the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) algorithm [Bro70, Fle70, Gol70, Sha70] which requires $\mathcal{O}(np + p^2)$ per-iteration cost [Bis95, Nes04].

An alternative approach is to use *sub-sampling* techniques, where scaling matrix $\mathbf{Q}^t$ is based on randomly selected set of data points [Mar10, BCNN11, VP12]. Sub-sampling is widely used in the first order methods, but is not as well studied for approximating the scaling matrix. In particular, theoretical guarantees are still missing.

A key challenge is that the sub-sampled Hessian is close to the actual Hessian along the directions corresponding to large eigenvalues (large curvature directions in $f(\theta)$), but is a poor approximation in the directions corresponding to small eigenvalues (flatter directions in $f(\theta)$). In order to overcome this problem, we use low-rank approximation. More precisely, we treat all the eigenvalues below the r-th as if they were equal to the $(r+1)$-th. This yields the desired stability with respect to the sub-sample: we call our algorithm NewSamp. In this paper, we establish the following:

1. NewSamp has a composite convergence rate: quadratic at start and linear near the minimizer, as illustrated in Figure 1. Formally, we prove a bound of the form

   $$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2 + \xi_2^t \|\hat{\theta}^t - \theta_*\|_2^2$$

   with coefficient that are explicitly given (and are computable from data).

2. The asymptiotic behavior of the linear convergence coefficient is $\lim_{t\to\infty} \xi_1^t = 1 - (\lambda_p^*/\lambda_{r+1}^*) + \delta$, for δ small. The condition number $(\lambda_1^*/\lambda_p^*)$ which controls the convergence of GD, has been replaced by the milder $(\lambda_{r+1}^*/\lambda_p^*)$. For datasets with strong spectral features, this can be a large improvement, as shown in Figure 1.

3. The above results are achived without tuning the step-size, in particular, by setting $\eta_t = 1$.

4. The complexity per iteration of NewSamp is $\mathcal{O}(np + |S|p^2)$ with $|S|$ the sample size.

5. Our theoretical results can be used to obtain convergence rates of previously proposed sub-sampling algorithms.

We demonstrate the performance of NewSamp on four datasets, and compare it to the well-known optimization methods.

The rest of the paper is organized as follows: Section 1.1 surveys the related work. In Section 2, we describe the proposed algorithm and provide the intuition behind it. Next, we present our theoretical results in Section 3, i.e., convergence rates corresponding to different sub-sampling schemes, followed by a discussion on how to choose the algorithm parameters. Two applications of the algorithm are discussed in Section 4. We compare our algorithm with several existing methods on various datasets in Section 5. Finally, in Section 6, we conclude with a brief discussion.

1.1 Related Work

Even a synthetic review of optimization algorithms for large-scale machine learning would go beyond the page limits of this paper. Here, we emphasize that the method of choice depends crucially on the amount of data to be used, and their dimensionality (i.e., respectively, on the parameters n and p). In this paper, we focus on a regime in which p is large but not so large as to make matrix manipulations (of order p^2 to p^3) impossible. Also n is large but not so large as to make batch gradient computation (of order np) prohibitive. On the other hand, our aim is to avoid $\mathcal{O}(np^2)$ calculations required by standard Newton method. Examples of this regime are given in Section 4.

In contrast, online algorithms are the option of choice for very large n since the computation per update is independent of n. In the case of *Stochastic Gradient Descent* (SGD), the descent direction is formed by a randomly selected gradient [RM51]. Improvements to SGD have been developed by incorporating the previous gradient directions in the current update [SRB13, SHRY13, Bot10, DHS11].

Batch algorithms, on the other hand, can achieve faster convergence and exploit second order information. They are competitive for intermediate n. Several methods in this category aim at quadratic, or at least super-linear convergence rates. In particular, Quasi-Newton methods have proven effective [Bis95, Nes04]. Another approach towards the same goal is to utilize sub-sampling to form an approximate Hessian [Mar10, BCNN11, VP12, QRTF15, EM15, Erd15a]. If the sub-sampled Hessian is close to the true Hessian, these methods can approach NM in terms of convergence rate, nevertheless, they enjoy much smaller complexity per update. No convergence rate analysis is available for these methods: this analysis is the main contribution of our paper. To the best of our knowledge, the best result in this direction is proven in [BCNN11] that estabilishes asymptotic convergence without quantitative bounds (exploiting general theory from [GNS09]).

Further improvements have been suggested either by utilizing *Conjugate Gradient* (CG) methods and/or using Krylov sub-spaces [Mar10, BCNN11, VP12]. Sub-sampling can be also used to obtain an approximate solution, if an exact solution is not required [DLFU13]. Lastly, there are various hybrid algorithms that combine two or more techniques to gain improvement. Examples include, sub-sampling and Quasi-Newton [SYG07, SDPG13, BHNS14], SGD and GD [FS12], NGD and NM [RF10], NGD and low-rank approximation [RaMB08].

Algorithm 1 NewSamp

Input: $\hat{\theta}^0, r, \epsilon, \{\eta_t, |S_t|\}_t, t = 0$.

1. **Define:** $\mathcal{P}_{\mathcal{C}}(\theta) = \text{argmin}_{\theta' \in \mathcal{C}} \|\theta - \theta'\|_2$ is the Euclidean projection onto $\mathcal{C}$,
   $[\mathbf{U}_k, \mathbf{\Lambda}_k] = \text{TruncatedSVD}_k(\mathbf{H})$ is the rank-k truncated SVD of $\mathbf{H}$ with $(\mathbf{\Lambda}_k)_{ii} = \lambda_i$.
2. **while** $\|\hat{\theta}^{t+1} - \hat{\theta}^t\|_2 \leq \epsilon$ **do**
   Sub-sample a set of indices $S_t \subset [n]$.
   Let $\mathbf{H}_{S_t} = \frac{1}{|S_t|} \sum_{i \in S_t} \nabla^2_\theta f_i(\hat{\theta}^t)$, and $[\mathbf{U}_{r+1}, \mathbf{\Lambda}_{r+1}] = \text{TruncatedSVD}_{r+1}(\mathbf{H}_{S_t})$,
   $\mathbf{Q}^t = \lambda_{r+1}^{-1} \mathbf{I}_p + \mathbf{U}_r \left(\mathbf{\Lambda}_r^{-1} - \lambda_{r+1}^{-1} \mathbf{I}_r \right) \mathbf{U}_r^T$,
   $\hat{\theta}^{t+1} = \mathcal{P}_{\mathcal{C}} \left(\hat{\theta}^t - \eta_t \mathbf{Q}^t \nabla_\theta f(\hat{\theta}^t) \right)$,
   $t \leftarrow t + 1$.
3. **end while**

Output: $\hat{\theta}^t$.

2 NewSamp : A Newton method via sub-sampling and eigenvalue thresholding

In the regime we consider, $n \gg p \gg 1$, there are two main drawbacks associated with the classical second order methods such as Newton's method. The predominant issue in this regime is the computation of the Hessian matrix, which requires $\mathcal{O}(np^2)$ operations, and the other issue is finding the inverse of the Hessian, which requires $\mathcal{O}(p^3)$ computation. Sub-sampling is an effective and efficient way of addressing the first issue, by forming an approximate Hessian to exploit curvature information. Recent empirical studies show that sub-sampling the Hessian provides significant improvement in terms of computational cost, yet preserves the fast convergence rate of second order methods [Mar10, VP12, Erd15b]. If a uniform sub-sample is used, the sub-sampled Hessian will be a random matrix with expected value at the true Hessian, which can be considered as a sample estimator to the mean. Recent advances in statistics have shown that the performance of various estimators can be significantly improved by simple procedures such as *shrinkage* and/or *thresholding* [CCS10, DGJ13, GD14, GD14]. To this extent, we use a specialized low-rank approximation as the important second order information is generally contained in the largest few eigenvalues/vectors of the Hessian. We will see in Section 3, how this procedure provides faster convergence rates compared to the bare sub-sampling methods.

NewSamp is presented as Algorithm 1. At iteration step t, the sub-sampled set of indices, its size and the corresponding sub-sampled Hessian is denoted by S_t, $|S_t|$ and $\mathbf{H}_{S_t}$, respectively. Assuming that the functions f_i's are convex, eigenvalues of the symmetric matrix $\mathbf{H}_{S_t}$ are non-negative. Therefore, singular value (SVD) and eigenvalue decompositions coincide. The operation $\text{TruncatedSVD}_k(\mathbf{H}_{S_t}) = [\mathbf{U}_k, \mathbf{\Lambda}_k]$ is the best rank-k approximation, i.e., takes $\mathbf{H}_{S_t}$ as input and returns the largest k eigenvalues in the diagonal matrix $\mathbf{\Lambda}_k \in \mathbb{R}^{k \times k}$ with the corresponding k eigenvectors $\mathbf{U}_k \in \mathbb{R}^{p \times k}$. This procedure requires $\mathcal{O}(kp^2)$ computation using a standard method, though there are faster randomized algorithms which provide accurate approximations to the truncated SVD problem with much less computational cost [HMT11]. To construct the curvature matrix $[\mathbf{Q}^t]^{-1}$, instead of using the basic rank-r approximation, we fill its 0 eigenvalues with the $(r+1)$-th eigenvalue

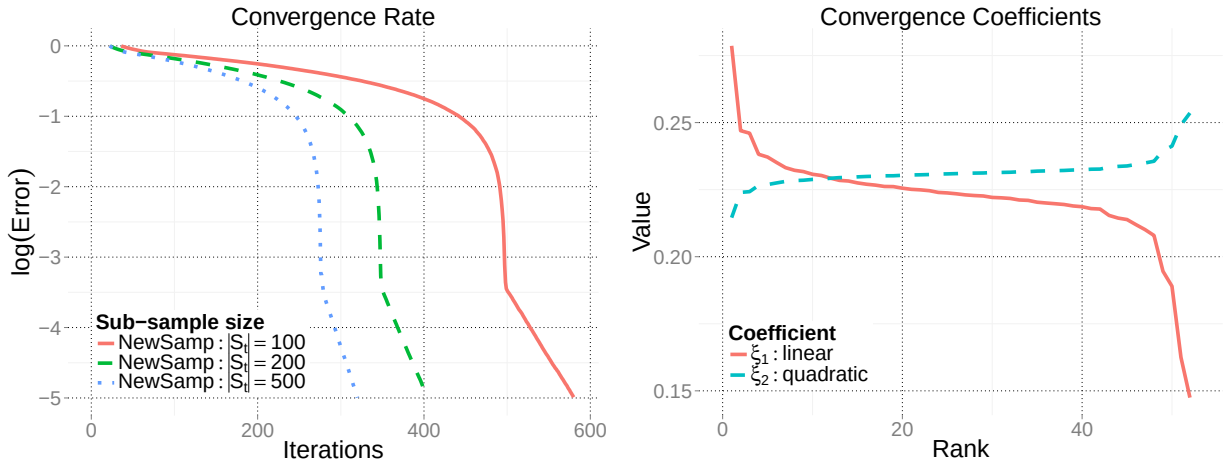

Figure 1: Left plot demonstrates convergence rate of NewSamp , which starts with a quadratic rate and transitions into linear convergence near the true minimizer. The right plot shows the effect of eigenvalue thresholding on the convergence coefficients. x-axis shows the number of kept eigenvalues. Plots are obtained using *Covertype* dataset.

of the sub-sampled Hessian which is the largest eigenvalue below the threshold. If we compute a truncated SVD with $k = r + 1$ and $(\mathbf{\Lambda}_k)_{ii} = \lambda_i$, the described operation can be formulated as the following,

$$\mathbf{Q}^t = \lambda_{r+1}^{-1}\mathbf{I}_p + \mathbf{U}_r \left(\mathbf{\Lambda}_r^{-1} - \lambda_{r+1}^{-1}\mathbf{I}_r\right) \mathbf{U}_r^T, \tag{2.1}$$

which is simply the sum of a scaled identity matrix and a rank-r matrix. Note that the low-rank approximation that is suggested to improve the curvature estimation has been further utilized to reduce the cost of computing the inverse matrix. Final per-iteration cost of NewSamp will be $\mathcal{O}\left(np + (|S_t| + r)p^2\right) \approx \mathcal{O}\left(np + |S_t|p^2\right)$. NewSamp takes the parameters $\{\eta_t, |S_t|\}_t$ and r as inputs. We discuss in Section 3.4, how to choose these parameters near-optimally, based on the theory we develop in Section 3.

Operator $\mathcal{P}_\mathcal{C}$ projects the current iterate to the feasible set $\mathcal{C}$ using Euclidean projection. Throughout, we assume that this projection can be done efficiently. In general, most unconstrained optimization problems do not require this step, and can be omitted. The purpose of projected iterations in our algorithm is mostly theoretical, and will be clear in Section 3.

By the construction of $\mathbf{Q}^t$, NewSamp will always be a descent algorithm. It enjoys a quadratic convergence rate at start which transitions into a linear rate in the neighborhood of the minimizer. This behavior can be observed in Figure 1. The left plot in Figure 1 shows the convergence behavior of NewSamp over different sub-sample sizes. We observe that large sub-samples result in better convergence rates as expected. As the sub-sample size increases, slope of the linear phase decreases, getting closer to that of quadratic phase at the transition point. This phenomenon will be explained in detail in Section 3, by Theorems 3.2 and 3.4. The right plot in Figure 1 demonstrates how the coefficients of linear and quadratic phases depend on the thresholded rank. Note that the coefficient of the quadratic phase increases with the rank threshold, whereas for the linear phase, relation is reversed.

3 Theoretical results

In this section, we provide the convergence analysis of NewSamp based on two different sub-sampling schemes:

S1: **Independent sub-sampling**: At each iteration t, S_t is uniformly sampled from $[n] = \{1, 2, ..., n\}$, independently from the sets $\{S_\tau\}_{\tau<t}$, with or without replacement.

S2: **Sequentially dependent sub-sampling**: At each iteration t, S_t is sampled from $[n]$, based on a distribution which might depend on the previous sets $\{S_\tau\}_{\tau<t}$, but not on any randomness in the data.

The first sub-sampling scheme is simple and commonly used in optimization. One drawback is that the sub-sampled set at the current iteration is independent of the previous sub-samples, hence does not consider which of the samples were previously used to form the approximate curvature information. In order to prevent cycles and obtain better performance near the optimum, one might want to increase the sample size as the iteration advances [Mar10], including previously unused samples. This process results in a sequence of dependent sub-samples which falls into the sub-sampling scheme S2. In our theoretical analysis, we make the following assumptions:

Assumption 1 (Lipschitz continuity)**.** *For any subset $S \subset [n]$, there exists a constant $M_{|S|}$ depending on the size of S, such that $\forall \theta, \theta' \in \mathcal{C}$,*

$$\left\|\mathbf{H}_S(\theta) - \mathbf{H}_S(\theta')\right\|_2 \leq M_{|S|} \ \|\theta - \theta'\|_2.$$

Assumption 2 (Bounded Hessian)**.** *$\forall i = 1, 2, ..., n$, the Hessian of the function $f_i(\theta)$, $\boldsymbol{\nabla}^2_\theta f_i(\theta)$, is upper bounded by an absolute constant K, i.e.,*

$$\max_{i\leq n} \left\|\boldsymbol{\nabla}^2_\theta f_i(\theta)\right\|_2 \leq K.$$

3.1 Independent sub-sampling

In this section, we assume that $S_t \subset [n]$ is sampled according to the sub-sampling scheme S1. In fact, many stochastic algorithms assume that S_t is a uniform subset of $[n]$, because in this case the sub-sampled Hessian is an unbiased estimator of the full Hessian. That is, $\forall \theta \in \mathcal{C}$, $\mathbb{E}\left[\mathbf{H}_{S_t}(\theta)\right] = \mathbf{H}_{[n]}(\theta)$, where the expectation is over the randomness in S_t. We next show that for any scaling matrix $\mathbf{Q}^t$ that is formed by the sub-samples S_t, iterations of the form Eq. (1.2) will have a composite convergence rate, i.e., combination of a linear and a quadratic phases.

Lemma 3.1. *Assume that the parameter set $\mathcal{C}$ is convex and $S_t \subset [n]$ is based on sub-sampling scheme S1. Further, let the Assumptions 1 and 2 hold and $\theta_* \in \mathcal{C}$. Then, for an absolute constant $c > 0$, with probability at least $1 - 2/p$, the updates of the form Eq. (1.2) satisfy*

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2 + \xi_2^t \|\hat{\theta}^t - \theta_*\|_2^2,$$

for coefficients ξ_1^t and ξ_2^t defined as

$$\xi_1^t = \left\|I - \eta_t \mathbf{Q}^t \mathbf{H}_{S_t}(\hat{\theta}^t)\right\|_2 + \eta_t c K \left\|\mathbf{Q}^t\right\|_2 \sqrt{\frac{\log(p)}{|S_t|}}, \qquad \xi_2^t = \eta_t \frac{M_n}{2} \left\|\mathbf{Q}^t\right\|_2.$$

Remark 1. *If the initial point $\hat{\theta}^0$ is close to θ_*, the algorithm will start with a quadratic rate of convergence which will transform into linear rate later in the close neighborhood of the optimum.*

The above lemma holds for any matrix $\mathbf{Q}^t$. In particular, if we choose $\mathbf{Q}^t = \mathbf{H}_{S_t}^{-1}$, we obtain a bound for the simple sub-sampled Hessian method. In this case, the coefficients ξ_1^t and ξ_2^t depend on $\|\mathbf{Q}^t\|_2 = 1/\lambda_p^t$ where λ_p^t is the smallest eigenvalue of the sub-sampled Hessian. Note that λ_p^t can be arbitrarily small which might blow up both of the coefficients. In the following, we will see how NewSamp remedies this issue.

Theorem 3.2. *Let the assumptions in Lemma 3.1 hold. Denote by λ_i^t, the i-th eigenvalue of $\mathbf{H}_{S_t}(\hat{\theta}^t)$ where $\hat{\theta}^t$ is given by NewSamp at iteration step t. If the step size satisfies*

$$\eta_t \leq \frac{2}{1 + \lambda_p^t/\lambda_{r+1}^t}, \tag{3.1}$$

then we have, with probability at least $1 - 2/p$,

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2 + \xi_2^t \|\hat{\theta}^t - \theta_*\|_2^2,$$

for an absolute constant $c > 0$, for the coefficients ξ_1^t and ξ_2^t are defined as

$$\xi_1^t = 1 - \eta_t \frac{\lambda_p^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t} \sqrt{\frac{\log(p)}{|S_t|}}, \qquad \xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t}.$$

NewSamp has a composite convergence rate where ξ_1^t and ξ_2^t are the coefficients of the linear and the quadratic terms, respectively (See the right plot in Figure 1). We observe that the sub-sampling size has a significant effect on the linear term, whereas the quadratic term is governed by the Lipschitz constant. We emphasize that the case $\eta_t = 1$ is feasible for the conditions of Theorem 3.2. In the case of quadratic functions, since the Lipschitz constant is 0 , we obtain $\xi_2^t = 0$ and the algorithm converges linearly. Following corollary summarizes this case.

Corollary 3.3 (Quadratic functions)**.** *Let the assumptions of Theorem 3.2 hold. Further, assume that $\forall i \in [n]$, the functions $\theta : \mathbb{R}^p \to f_i(\theta)$ are quadratic. Then, for $\hat{\theta}^t$ given by NewSamp at iteration step t, for the coefficient ξ_1^t defined as in Theorem 3.2, with probability at least $1 - 2/p$, we have*

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2. \tag{3.2}$$

3.2 Sequentially dependent sub-sampling

Here, we assume that the sub-sampling scheme S2 is used to generate $\{S_\tau\}_{\tau \geq 1}$. Distribution of sub-sampled sets may depend on each other, but not on any randomness in the dataset. Examples include fixed sub-samples as well as sub-samples of increasing size, sequentially covering unused data. In addition to Assumptions 1-2, we assume the following.

Assumption 3 (i.i.d. observations)**.** *Let $z_1, z_2, ..., z_n \in \mathrm{Z}$ be i.i.d. observations from a distribution $\mathcal{D}$. For a fixed $\theta \in \mathbb{R}^p$ and $\forall i \in [n]$, we assume that the functions $\{f_i\}_{i=1}^n$ satisfy $f_i(\theta) = \varphi(z_i, \theta)$, for some function $\varphi : \mathrm{Z} \times \mathbb{R}^p \to \mathbb{R}$.*

Most statistical learning algorithms can be formulated as above, e.g., in classification problems, one has access to i.i.d. samples $\{(y_i, x_i)\}_{i=1}^n$ where y_i and x_i denote the class label and the covariate, and φ measures the classification error (See Section 4 for examples). For the sub-sampling scheme S2, an analogue of Lemma 3.1 is stated in Appendix as Lemma A.1, which immediately leads to the following theorem.

Theorem 3.4. *Assume that the parameter set $\mathcal{C}$ is convex and $S_t \subset [n]$ is based on the sub-sampling scheme S2. Further, let the Assumptions 1, 2 and 3 hold, almost surely. Conditioned on the event $\mathcal{E} = \{\theta_* \in \mathcal{C}\}$, if the step size satisfies Eq. 3.1, then for $\hat{\theta}^t$ given by NewSamp at iteration t, with probability at least $1 - c_{\mathcal{E}}\, e^{-p}$ for $c_{\mathcal{E}} = c/\mathbb{P}(\mathcal{E})$, we have*

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2 + \xi_2^t \|\hat{\theta}^t - \theta_*\|_2^2,$$

for the coefficients ξ_1^t and ξ_2^t defined as

$$\xi_1^t = 1 - \eta_t \frac{\lambda_p^t}{\lambda_{r+1}^t} + \eta_t \frac{c' K}{\lambda_{r+1}^t} \sqrt{\frac{p}{|S_t|} \log\left(\frac{\operatorname{diam}(\mathcal{C})^2 \left(M_n + M_{|S_t|}\right)^2 |S_t|}{K^2} \right)}, \qquad \xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t},$$

where $c, c' > 0$ are absolute constants and λ_i^t denotes the i-th eigenvalue of $\mathbf{H}_{S_t}(\hat{\theta}^t)$.

Compared to the Theorem 3.2, we observe that the coefficient of the quadratic term does not change. This is due to Assumption 1. However, the bound on the linear term is worse, since we use the uniform bound over the convex parameter set $\mathcal{C}$. The same order of magnitude is also observed by [Erd15b], which relies on a similar proof technique. Similar to Corollary 3.3, we have the following result for the quadratic functions.

Corollary 3.5 (Quadratic functions)**.** *Let the assumptions of Theorem 3.4 hold. Further assume that $\forall i \in [n]$, the functions $\theta \to f_i(\theta)$ are quadratic. Then, conditioned on the event $\mathcal{E}$, with probability at least $1 - c_{\mathcal{E}}\, e^{-p}$, NewSamp iterates satisfy*

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2,$$

for coefficient ξ_1^t defined as in Theorem 3.4.

3.3 Dependence of coefficients on t and convergence guarantees

The coefficients ξ_1^t and ξ_2^t depend on the iteration step which is an undesirable aspect of the above results. However, these constants can be well approximated by their analogues ξ_1^* and ξ_2^* evaluated at the optimum which are defined by simply replacing λ_j^t with λ_j^* in their definition, where the latter is the j-th eigenvalue of full-Hessian at θ_*. For the sake of simplicity, we only consider the case where the functions $\theta \to f_i(\theta)$ are quadratic.

Theorem 3.6. *Assume that the functions $f_i(\theta)$ are quadratic, S_t is based on scheme S1 and $\eta_t = 1$. Let the full Hessian at θ_* be lower bounded by a constant k. Then for sufficiently large $|S_t|$, we have, with probability $1 - 2/p$*

$$\left|\xi_1^t - \xi_1^*\right| \leq \frac{c_1 K \sqrt{\log(p)/|S_t|}}{k\big(k - c_2 K \sqrt{\log(p)/|S_t|}\big)} \coloneqq \delta,$$

for some absolute constants c_1, c_2.

Theorem 3.6 implies that, when the sub-sampling size is sufficiently large, ξ_1^t will concentrate around ξ_1^*. Generalizing the above theorem to non-quadratic functions is straightforward, in which case, one would get additional terms involving the difference $\|\hat{\theta}^t - \theta_*\|_2$. In the case of scheme S2, if one uses fixed sub-samples, i.e., $\forall t$, $S_t = S$, then the coefficient ξ_1^t does not depend on t. The following corollary gives a sufficient condition for convergence. A detailed discussion on the number of iterations until convergence and further local convergence properties can be found in Appendix B.

Corollary 3.7. *Assume that ξ_1^t and ξ_2^t are well-approximated by ξ_1^* and ξ_2^* with an error bound of δ, i.e., $\xi_i^t \leq \xi_i^* + \delta$ for $i = 1, 2$, as in Theorem 3.6. For the initial point $\hat{\theta}^0$, a sufficient condition for convergence is*

$$\|\hat{\theta}^0 - \theta_*\|_2 < \frac{1 - \xi_1^* - \delta}{\xi_2^* + \delta}.$$

3.4 Choosing the algorithm parameters

Algorithm parameters play a crucial role in most optimization methods. Based on the theoretical results from previous sections, we discuss procedures to choose the optimal values for the step size η_t, sub-sample size $|S_t|$ and rank threshold.

- *Step size:* For the step size of NewSamp at iteration t, we suggest

$$\eta_t(\gamma) = \frac{2}{1 + \lambda_p^t/\lambda_{r+1}^t + \gamma}. \tag{3.3}$$

  where $\gamma = \mathcal{O}(\log(p)/|S_t|)$. Note that $\eta_t(0)$ is the upper bound in Theorems 3.2 and 3.4 and it minimizes the first component of ξ_1^t. The other terms in ξ_1^t and ξ_2^t linearly depend on η_t. To compensate for that, we shrink $\eta_t(0)$ towards 1. Contrary to most algorithms, optimal step size of NewSamp is larger than 1. See Appendix C for a rigorous derivation of Eq. 3.3.

- *Sample size:* By Theorem 3.2, a sub-sample of size $\mathcal{O}((K/\lambda_p^*)^2 \log(p))$ should be sufficient to obtain a small coefficient for the linear phase. Also note that sub-sample size $|S_t|$ scales quadratically with the condition number.

- *Rank threshold:* For a full-Hessian with effective rank R (trace divided by the largest eigenvalue), it suffices to use $\mathcal{O}(R \log(p))$ samples [Ver10, Ver12]. Effective rank is upper bounded by the dimension p. Hence, one can use $p \log(p)$ samples to approximate the full-Hessian and choose a rank threshold which retains the important curvature information.

4 Examples

4.1 Generalized Linear Models

Finding the maximum likelihood estimator in Generalized Linear Models (GLMs) is equivalent to minimizing the negative log-likelihood $f(\theta)$,

$$\underset{\theta}{\text{minimize}}\ f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\Phi(\langle x_i, \theta \rangle) - y_i \langle x_i, \theta \rangle\right], \tag{4.1}$$

where Φ is the *cumulant generating function*, $y_i \in \mathbb{R}$ denotes the observations, $x_i \in \mathbb{R}^p$ denotes the rows of design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, and $\theta \in \mathbb{R}^p$ is the coefficient vector. Note that this formulation only considers GLMs with canonical links. Here, $\langle x, \theta \rangle$ denotes the inner product between the vectors x, θ. The function Φ defines the type of GLM. Well known examples include ordinary least squares (OLS) with $\Phi(z) = z^2$, logistic regression (LR) with $\Phi(z) = \log(1 + e^z)$, and Poisson regression (PR) with $\Phi(z) = e^z$.

The gradient and the Hessian of the above function can be written as:

$$\nabla_\theta f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\Phi^{(1)}(\langle x_i, \theta \rangle) x_i - y_i x_i \right], \qquad \boldsymbol{\nabla}^2_\theta f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \Phi^{(2)}(\langle x_i, \theta \rangle) x_i x_i^T. \tag{4.2}$$

We note that the Hessian of the GLM problem is always positive definite. This is because the second derivative of the cumulant generating function is simply the variance of the observations. Using the results from Section 3, we perform a convergence analysis of our algorithm on a GLM problem.

Corollary 4.1. *Let $S_t \subset [n]$ be a uniform sub-sample, and $\mathcal{C}$ be a convex parameter set. Assume that the second derivative of the cumulant generating function, $\Phi^{(2)}$ is bounded by 1, and it is Lipschitz continuous with Lipschitz constant L. Further, assume that the covariates are contained in a ball of radius $\sqrt{R_x}$, i.e. $\max_{i \in [n]} \|x_i\|_2 \leq \sqrt{R_x}$. Then, for $\hat{\theta}^t$ given by NewSamp with constant step size $\eta_t = 1$ at iteration t, with probability at least $1 - 2/p$, we have*

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2 + \xi_2^t \|\hat{\theta}^t - \theta_*\|_2^2,$$

for constants ξ_1^t and ξ_2^t defined as

$$\xi_1^t = 1 - \frac{\lambda_i^t}{\lambda_{r+1}^t} + \frac{c R_x}{\lambda_{r+1}^t} \sqrt{\frac{\log(p)}{|S_t|}}, \qquad \xi_2^t = \frac{L R_x^{3/2}}{2 \lambda_{r+1}^t},$$

where $c > 0$ is an absolute constant and λ_i^t is the ith eigenvalue of $\mathbf{H}_{S_t}(\hat{\theta}^t)$.

Proof of Corollary 4.1 can be found in Appendix A. Note that the bound on the second derivative is quite loose for Poisson regression due to exponentially fast growing cumulant generating function.

4.2 Support Vector Machines

A linear Support Vector Machine (SVM) provides a *separating hyperplane* which maximizes the *margin*, i.e., the distance between the hyperplane and the support vectors. Although the vast majority of the literature focuses on the dual problem [Vap98, SS02], SVMs can be trained using the primal as well. Since the dual problem does not scale well with the number of data points (some approaches get $\mathcal{O}(n^3)$ complexity, [WG11]), the primal might be better-suited for optimization of linear SVMs [KD05, Cha07].

The primal problem for the linear SVM can be written as

$$\underset{\theta \in \mathcal{C}}{\text{minimize}} \; f(\theta) = \frac{1}{2} \|\theta\|_2^2 + \frac{1}{2} C \sum_{i=1}^{n} \ell(y_i, \langle \theta, x_i \rangle) \tag{4.3}$$

where (y_i, x_i) denote the data samples, θ defines the separating hyperplane, $C > 0$ and ℓ could be any loss function. The most commonly used loss functions include *Hinge-p loss*, *Huber loss* and their

smoothed versions [Cha07]. Smoothing or approximating such losses with more stable functions is sometimes crucial in optimization. In the case of NewSamp which requires the loss function to be twice differentiable (almost everywhere), we suggest either smoothed Huber loss, i.e.,

$$\ell(y, \langle \theta, x \rangle) = \begin{cases} 0, & \text{if } y\langle \theta, x \rangle > 3/2, \\ \frac{(3/2 - y\langle \theta, x \rangle)^2}{2}, & \text{if } |1 - y\langle \theta, x \rangle| \leq 1/2, \\ 1 - y\langle \theta, x \rangle, & \text{otherwise.} \end{cases}$$

or Hinge-2 loss, i.e.,

$$\ell(y, \langle \theta, x \rangle) = \max\{0, 1 - y\langle \theta, x \rangle\}^2 .$$

For the sake of simplicity, we will focus on Hinge-2 loss. Denote by SV_t, the set of indices of all the support vectors at iteration t, i.e.,

$$SV_t = \{i : y_i \langle \theta^t, x_i \rangle < 1\}.$$

When the loss is set to be the Hinge-2 loss, the Hessian of the SVM problem, normalized by the number of support vectors, can be written as

$$\nabla^2_\theta f(\theta) = \frac{1}{|SV_t|}\Big\{\mathbf{I} + C \sum_{i \in SV_t} x_i x_i^T\Big\}.$$

When $|SV_t|$ is large, the problem falls into our setup and can be solved efficiently using NewSamp . Note that unlike the GLM setting, Lipschitz condition of our Theorems do not apply here. However, we empirically demonstrate that NewSamp works regardless of such assumptions.

5 Experiments

In this section, we validate the performance of NewSamp through extensive numerical studies. We experimented on two optimization problems, namely, *Logistic Regression* (LR) and *Support Vector Machines* (SVM) with quadratic loss. LR minimizes Eq. 4.1 for the logistic function, whereas SVM minimizes Eq. 4.3 for the Hinge-2 loss.

In the following, we briefly describe the algorithms that are used in the experiments:

1. *Gradient Descent* (GD), at each iteration, takes a step proportional to negative of the full gradient evaluated at the current iterate. Under certain regularity conditions, GD exhibits a linear convergence rate.

2. *Accelerated Gradient Descent* (AGD) is proposed by Nesterov [Nes83], which improves over the gradient descent by using a momentum term. Performance of AGD strongly depends of the smoothness of the function f and decreasing step size adjustments may be necessary for convergence.

3. *Newton's Method* (NM) achieves a quadratic convergence rate by utilizing the inverse Hessian evaluated at the current iterate. However, the computation of Hessian makes it impractical for large-scale datasets.

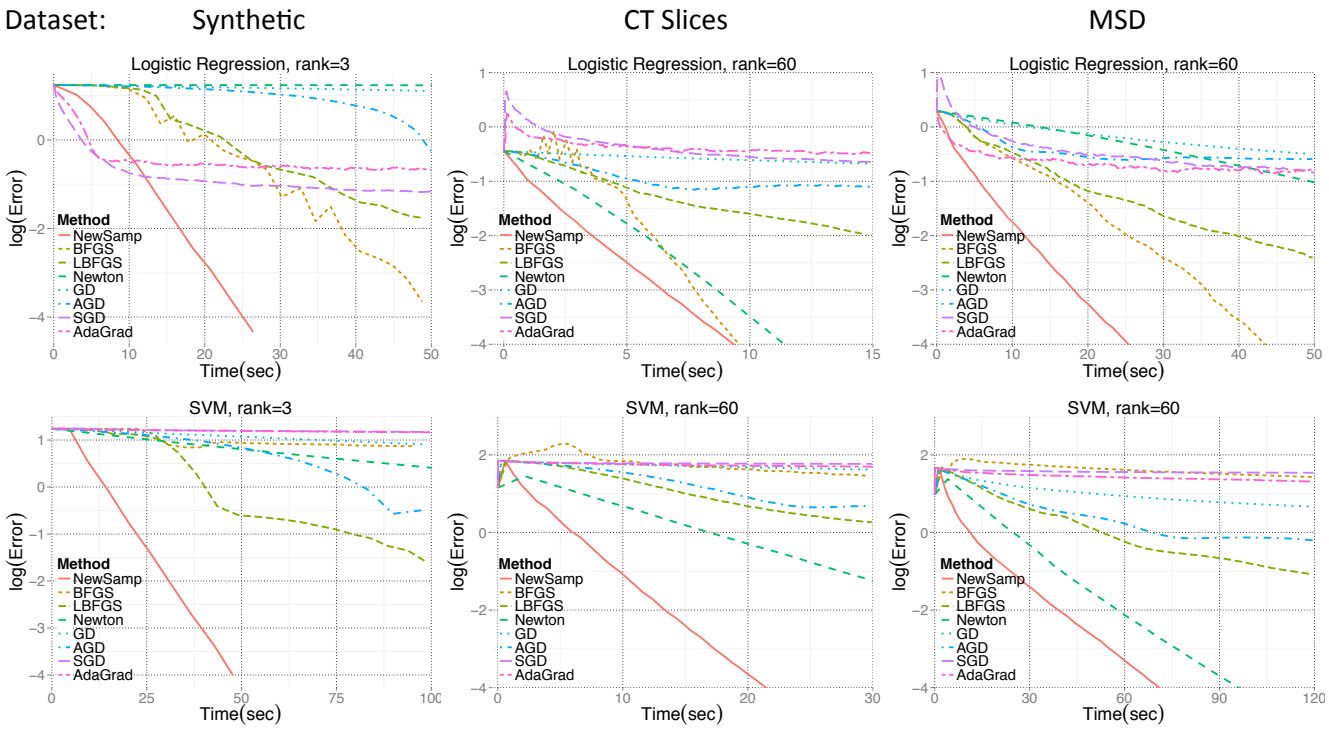

Figure 2: Performance of various optimization methods on different datasets. NewSamp is represented with red color .

4. *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) is the most popular and stable Quasi-Newton method. Scaling matrix is formed by accumulating the information from iterates and gradients, satisfying *Quasi-Newton rule*. The convergence rate is locally super-linear and per-iteration cost is comparable to first order methods.

5. *Limited Memory BFGS* (L-BFGS) is a variant of BFGS, which uses only the recent iterates and gradients to form the approximate Hessian, providing significant improvement in terms of memory usage.

6. *Stochastic Gradient Descent* (SGD) is a simplified version of GD where, at each iteration, instead of the full gradient, a randomly selected gradient is used. Per-iteration cost is independent of n, yet the convergence rate is significantly slower compared to batch algorithms. We follow the guidelines of [Bot10, SHRY13] for the step size,, i.e.,

$$\gamma_t = \frac{\gamma}{1 + t/c},$$

for constants $\gamma, c > 0$.

7. *Adaptive Gradient Scaling* (AdaGrad) is an online algorithm which uses an adaptive learning rate based on the previous gradients. AdaGrad significantly improves the performance and stability of SGD [DHS11]. This is achieved by scaling each entry of gradient differently. , i.e.,

at iteration step t, step size for the j-th coordinate is

$$(\gamma_t)_j = \frac{\gamma}{\sqrt{\delta + \sum_{\tau=1}^{t} (\nabla_\theta f(\hat{\theta}^t))_j}},$$

for constants $\delta, \gamma > 0$.

For each of the batch algorithms, we used constant step size, and for all the algorithms, we choose the step size that provides the fastest convergence. For the stochastic algorithms, we optimized over the parameters that define the step size. Parameters of NewSamp are selected following the guidelines described in Section 3.4.

We experimented over various datasets that are given in Table 1. The real datasets are downloaded from the UCI repository [Lic13]. Each dataset consists of a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the corresponding observations (classes) $y \in \mathbb{R}^n$. Synthetic data is generated through a multivariate Gaussian distribution with a randomly generated covariance matrix. As a methodological choice, we selected moderate values of p, for which Newton's Method can still be implemented, and nevertheless we can demonstrate an improvement. For larger values of p, comparison is even more favorable to our approach.

The effects of sub-sampling size $|S_t|$ and rank threshold are demonstrated in Figure 1. A thorough comparison of the aforementioned optimization techniques is presented in Figure 2. In the case of LR, we observe that stochastic algorithms enjoy fast convergence at start, but slows down later as they get close to the true minimizer. The algorithm that comes close to NewSamp in terms of performance is BFGS. In the case of SVM, Newton's method is the closest algorithm to NewSamp, yet in all scenarios, NewSamp outperforms its competitors. Note that the global convergence of BFGS is not better than that of GD [Nes04]. The condition for super-linear rate is $\sum_t \|\theta^t - \theta_*\|_2 < \infty$ for which, an initial point close to the optimum is required [DM77]. This condition can be rarely satisfied in practice, which also affects the performance of the other second order methods. For NewSamp , even though the rank thresholding provides a certain level of robustness, we observed that the choice of a good starting point is still an important factor. Details about Figure 2 can be found in Table 3 in Appendix. For additional experiments and a detailed discussion, see Appendix D.

Dataset	n	p	r	Reference
CT slices	53500	386	60	[GKS+11]
Covertype	581012	54	20	[BD99]
MSD	515345	90	60	[BMEWL11]
Synthetic	500000	300	3	-

Table 1: Datasets used in the experiments.

6 Conclusion

In this paper, we proposed a sub-sampling based second order method utilizing low-rank Hessian estimation. The proposed method has the target regime $n \gg p$ and has $\mathcal{O}\left(np + |S|p^2\right)$ complexity per-iteration. We showed that the convergence rate of NewSamp is composite for two widely used

sub-sampling schemes, i.e., starts as quadratic convergence and transforms to linear convergence near the optimum. Convergence behavior under other sub-sampling schemes is an interesting line of research. Numerical experiments on both real and synthetic datasets demonstrate the performance of the proposed algorithm which we compared to the classical optimization methods.

Acknowledgments

We are grateful to Mohsen Bayati for stimulating conversations on the topic of this work. We would like to thank Robert M. Gower for carefully reading this manuscript and providing valuable feedback. A.M. was partially supported by NSF grants CCF-1319979 and DMS-1106627 and the AFOSR grant FA9550-13-1-0036.

References

[Ama98] Shun-Ichi Amari, *Natural gradient works efficiently in learning*, Neural computation **10** (1998), no. 2, 251–276.

[BCNN11] Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal, *On the use of stochastic hessian information in optimization methods for machine learning*, SIAM Journal on Optimization **21** (2011), no. 3, 977–995.

[BD99] Jock A Blackard and Denis J Dean, *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables*, Computers and electronics in agriculture **24** (1999), no. 3, 131–151.

[BHNS14] Richard H Byrd, SL Hansen, Jorge Nocedal, and Yoram Singer, *A stochastic quasi-newton method for large-scale optimization*, arXiv preprint arXiv:1401.7020 (2014).

[Bis95] Christopher M. Bishop, *Neural networks for pattern recognition*, Oxford University Press, Inc., NY, USA, 1995.

[BMEWL11] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere, *The million song dataset*, Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011), 2011.

[Bot10] Léon Bottou, *Large-scale machine learning with stochastic gradient descent*, Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[Bro70] Charles G Broyden, *The convergence of a class of double-rank minimization algorithms 2. the new algorithm*, IMA Journal of Applied Mathematics **6** (1970), no. 3, 222–231.

[BV04] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, New York, NY, USA, 2004.

[CCS10] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen, *A singular value thresholding algorithm for matrix completion*, SIAM Journal on Optimization **20** (2010), no. 4, 1956–1982.

[Cha07] Olivier Chapelle, *Training a support vector machine in the primal*, Neural Computation **19** (2007), no. 5, 1155–1178.

[DE15] Lee H Dicker and Murat A Erdogdu, *Flexible results for quadratic forms with applications to variance components estimation*, arXiv preprint arXiv:1509.04388 (2015).

[DGJ13] David L Donoho, Matan Gavish, and Iain M Johnstone, *Optimal shrinkage of eigenvalues in the spiked covariance model*, arXiv preprint arXiv:1311.0851 (2013).

[DHS11] John Duchi, Elad Hazan, and Yoram Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research **12** (2011), 2121–2159.

[DLFU13] Paramveer Dhillon, Yichao Lu, Dean P Foster, and Lyle Ungar, *New subsampling algorithms for fast least squares regression*, Advances in Neural Information Processing Systems 26, 2013, pp. 360–368.

[DM77] John E Dennis, Jr and Jorge J Moré, *Quasi-newton methods, motivation and theory*, SIAM review **19** (1977), 46–89.

[EM15] Murat A Erdogdu and Andrea Montanari, *Convergence rates of sub-sampled Newton methods*, Advances in Neural Information Processing Systems 29-(NIPS-15), 2015.

[Erd15a] Murat A Erdogdu, *Newton-Stein Method: A second order method for GLMs via Stein's lemma*, Advances in Neural Information Processing Systems 29-(NIPS-15), 2015.

[Erd15b] ———, *Newton-Stein Method: An optimization method for GLMs via Stein's Lemma*, arXiv preprint arXiv:1511.08895 (2015).

[Fle70] Roger Fletcher, *A new approach to variable metric algorithms*, The computer journal **13** (1970), no. 3, 317–322.

[FS12] Michael P Friedlander and Mark Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM Journal on Scientific Computing **34** (2012), no. 3, A1380–A1405.

[GD14] Matan Gavish and David L Donoho, *Optimal shrinkage of singular values*, arXiv:1405.7511 (2014).

[GKS+11] Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Sebastian Pölsterl, and Alexander Cavallaro, *2d image registration in ct images using radial image descriptors*, Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011, Springer, 2011, pp. 607–614.

[GN10] David Gross and Vincent Nesme, *Note on sampling without replacing from a finite collection of matrices*, arXiv preprint arXiv:1001.2738 (2010).

[GNS09] Igor Griva, Stephen G Nash, and Ariela Sofer, *Linear and nonlinear optimization*, Siam, 2009.

[Gol70] Donald Goldfarb, *A family of variable-metric methods derived by variational means*, Mathematics of computation **24** (1970), no. 109, 23–26.

[Gro11] David Gross, *Recovering low-rank matrices from few coefficients in any basis*, Information Theory, IEEE Transactions on **57** (2011), no. 3, 1548–1566.

[HMT11] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, no. 2, 217–288.

[KD05] S Sathiya Keerthi and Dennis DeCoste, *A modified finite newton method for fast solution of large scale linear svms*, Journal of Machine Learning Research, 2005, pp. 341–361.

[Lic13] M. Lichman, *UCI machine learning repository*, 2013.

[Mar10] James Martens, *Deep learning via hessian-free optimization*, Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 735–742.

[MJC+14] Lester Mackey, Michael I Jordan, Richard Y Chen, Brendan Farrell, Joel A Tropp, et al., *Matrix concentration inequalities via the method of exchangeable pairs*, The Annals of Probability **42** (2014), no. 3, 906–945.

[Nes83] Yurii Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence o (1/k2)*, Doklady AN SSSR, vol. 269, 1983, pp. 543–547.

[Nes04] ———, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer, 2004.

[QRTF15] Zheng Qu, Peter Richtárik, Martin Takáč, and Olivier Fercoq, *Sdna: Stochastic dual newton ascent for empirical risk minimization*, arXiv preprint arXiv:1502.02268 (2015).

[RaMB08] Nicolas L. Roux, Pierre antoine Manzagol, and Yoshua Bengio, *Topmoumoute online natural gradient algorithm*, Advances in Neural Information Processing Systems 20, 2008, pp. 849–856.

[RF10] Nicolas L Roux and Andrew W Fitzgibbon, *A fast natural newton method*, Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 623–630.

[RM51] Herbert Robbins and Sutton Monro, *A stochastic approximation method*, Annals of mathematical statistics (1951).

[SDPG13] Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli, *An adaptive low dimensional quasi-newton sum of functions optimizer*, arXiv preprint arXiv:1311.2115 (2013).

[Sha70] David F Shanno, *Conditioning of quasi-newton methods for function minimization*, Mathematics of computation **24** (1970), no. 111, 647–656.

[SHRY13] Alan Senior, Georg Heigold, Marc'Aurelio Ranzato, and Ke Yang, *An empirical study of learning rates in deep neural networks for speech recognition*, Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 6724–6728.

[SRB13] Mark Schmidt, Nicolas Le Roux, and Francis Bach, *Minimizing finite sums with the stochastic average gradient*, arXiv preprint arXiv:1309.2388 (2013).

[SS02] Bernhard Schölkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[SYG07] Nicol Schraudolph, Jin Yu, and Simon Günter, *A stochastic quasi-newton method for online convex optimization.*

[Tro12] Joel A Tropp, *User-friendly tail bounds for sums of random matrices*, Foundations of Computational Mathematics **12** (2012), no. 4, 389–434.

[Vap98] Vladimir Vapnik, *Statistical learning theory*, vol. 2, Wiley New York, 1998.

[VdVW96] Aad W Van der Vaart and Jon A Wellner, *Weak convergence*, Springer, 1996.

[Ver10] Roman Vershynin, *Introduction to the non-asymptotic analysis of random matrices*, arXiv:1011.3027 (2010).

[Ver12] ———, *How close is the sample covariance matrix to the actual covariance matrix?*, Journal of Theoretical Probability **25** (2012), no. 3, 655–686.

[VP12] Oriol Vinyals and Daniel Povey, *Krylov Subspace Descent for Deep Learning*, The 15th International Conference on Artificial Intelligence and Statistics - (AISTATS-12), 2012.

[WG11] Kristian Woodsend and Jacek Gondzio, *Exploiting separability in large-scale linear support vector machine training*, Computational Optimization and Applications **49** (2011), no. 2, 241–269.

A Proofs of Theorems and Lemmas

Proof of Lemma 3.1. We write,

$$\begin{aligned}\hat{\theta}^t - \theta_* - \eta_t \mathbf{Q}^t \nabla_\theta f(\hat{\theta}^t) &= \hat{\theta}^t - \theta_* - \eta_t \mathbf{Q}^t \int_0^1 \boldsymbol{\nabla}^2_\theta f(\theta_* + \tau(\hat{\theta}^t - \theta_*))(\hat{\theta}^t - \theta_*)\, d\tau, \\ &= \left(I - \eta_t \mathbf{Q}^t \int_0^1 \boldsymbol{\nabla}^2_\theta f(\theta_* + \tau(\hat{\theta}^t - \theta_*)) d\tau \right) (\hat{\theta}^t - \theta_*)\,.\end{aligned}$$

Since the projection $\mathcal{P}_\mathcal{C}$ in step 2 of NewSamp can only decrease the ℓ_2 distance, we obtain

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \left\| I - \eta_t \mathbf{Q}^t \int_0^1 \boldsymbol{\nabla}^2_\theta f(\theta_* + \tau(\hat{\theta}^t - \theta_*)) d\tau \right\|_2 \|\hat{\theta}^t - \theta_*\|_2.$$

Note that the first term on the right hand side governs the convergence behavior of the algorithm.

Next, for an index set $S \subset [n]$, define the matrix $\mathbf{H}_S(\theta)$ as

$$\mathbf{H}_S(\theta) = \frac{1}{|S|} \sum_{i \in S} \mathbf{H}_i(\theta)$$

where $|S|$ denotes the size of the set. Denote the integral in the above equation by $\widetilde{\mathbf{H}}$, that is,

$$\widetilde{\mathbf{H}} = \int_0^1 \boldsymbol{\nabla}^2_\theta f(\theta_* + \tau(\hat{\theta}^t - \theta_*)) d\tau.$$

By the triangle inequality, the governing term that determines the convergence rate can be bounded as

$$\begin{aligned}\left\| I - \eta_t \mathbf{Q}^t \widetilde{\mathbf{H}} \right\|_2 \leq \left\| I - \eta_t \mathbf{Q}^t \mathbf{H}_S(\hat{\theta}^t) \right\|_2 \\ + \eta_t \left\| \mathbf{Q}^t \right\|_2 \left\{ \left\| \mathbf{H}_S(\hat{\theta}^t) - \mathbf{H}_{[n]}(\hat{\theta}^t) \right\|_2 + \left\| \mathbf{H}_{[n]}(\hat{\theta}^t) - \widetilde{\mathbf{H}} \right\|_2 \right\},\end{aligned} \tag{A.1}$$

which holds, regardless of the choice of $\mathbf{Q}^t$.

In the following, we will use some matrix concentration results to bound the right hand side of Eq. (A.1). The result for sampling with replacement can be obtained by matrix Hoeffding's inequality given in [Tro12]. Note that this explicitly assumes that the samples are independent. For the concentration bounds under sampling without replacement (see i.e. [GN10, Gro11, MJC+14]), we will use the Operator-Bernstein inequality given in [GN10] which is provided in Section E as Lemma E.3 for convenience.

Using any indexing over the elements of sub-sample S, we denote the each element in S by s_i, i.e.,

$$S = \{s_1, s_2, ..., s_{|S|}\}.$$

For $\theta \in \mathcal{C}$, we define the centered Hessians, $\mathbf{W}_i(\theta)$ as

$$\mathbf{W}_i(\theta) = \mathbf{H}_{s_i}(\theta) - \mathbb{E}\left[\mathbf{H}_{s_i}(\theta)\right],$$

where the $\mathbb{E}\left[\mathbf{H}_{s_i}(\theta)\right]$ is just the full Hessian at θ.

By the Assumption (2), we have

$$\max_{i\le n}\|\mathbf{H}_i(\theta)\|_2 = \left\|\nabla^2_\theta f_i(\theta)\right\|_2 \le K, \tag{A.2}$$
$$\max_{i\le n}\|\mathbf{W}_i\|_2 \le 2K \coloneqq \gamma, \qquad \max_{i\le n}\left\|\mathbf{W}_i^2\right\|_2 \le 4K^2 \coloneqq \sigma^2.$$

Next, we apply the matrix Bernstein's inequality given in Lemma E.3. For $\epsilon \le 4K$, and $\theta \in \mathcal{C}$,

$$\mathbb{P}\left(\left\|\mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta)\right\|_2 > \epsilon\right) \le 2p\exp\left\{-\frac{\epsilon^2|S|}{16K^2}\right\}. \tag{A.3}$$

Therefore, to obtain a convergence rate of $\mathcal{O}(1/p)$, we let

$$\epsilon = C\sqrt{\frac{\log(p)}{|S|}},$$

where $C = 6K$ is sufficient. We also note that the condition on ϵ is trivially satisfied by our choice of ϵ in the target regime.

For the last term, we may write,

$$\begin{aligned}
\left\|\mathbf{H}_{[n]}(\hat{\theta}^t) - \widetilde{\mathbf{H}}\right\|_2 &= \left\|\mathbf{H}_{[n]}(\hat{\theta}^t) - \int_0^1 \nabla^2_\theta f(\theta_* + \tau(\hat{\theta}^t - \theta_*))d\tau\right\|_2, \\
&\le \int_0^1 \left\|\mathbf{H}_{[n]}(\hat{\theta}^t) - \nabla^2_\theta f(\theta_* + \tau(\hat{\theta}^t - \theta_*))\right\|_2 d\tau, \\
&\le \int_0^1 M_n(1-\tau)\|\hat{\theta}^t - \theta_*\|_2 d\tau, \\
&= \frac{M_n}{2}\|\hat{\theta}^t - \theta_*\|_2.
\end{aligned}$$

First inequality follows from the fact that norm of an integral is less than or equal to the integral of the norm. Second inequality follows from the Lipschitz property.

Combining the above results, we obtain the following for the governing term in Eq.(A.1): For some absolute constants $c, C > 0$, with probability at least $1 - 2/p$, we have

$$\left\|I - \eta_t\mathbf{Q}^t\mathbf{H}_{[n]}(\tilde{\theta}^t)\right\|_2 \le \left\|I - \eta_t\mathbf{Q}^t\mathbf{H}_S(\hat{\theta}^t)\right\|_2 + \eta_t\left\|\mathbf{Q}^t\right\|_2\left\{6K\sqrt{\frac{\log(p)}{|S|}} + \frac{M_n}{2}\ \|\hat{\theta}^t - \theta_*\|_2\right\}.$$

Hence, the proof is completed. □

Proof of Theorem 3.2. Using the definition of $\mathbf{Q}^t$ in NewSamp, we immediately obtain that

$$\left\|I - \eta_t\mathbf{Q}^t\mathbf{H}_{S_t}(\hat{\theta}^t)\right\|_2 = \max_{i>r}\left\{\left|1 - \eta_t\frac{\lambda_i^t}{\lambda_{r+1}^t}\right|\right\}, \tag{A.4}$$

and that $\left\|\mathbf{Q}^t\right\|_2 = 1/\lambda_{r+1}^t$. Then the proof follows from Lemma 3.1 and by the assumption on the step size. □

Lemma A.1. *Assume that the parameter set $\mathcal{C}$ is convex and $S_t \subset [n]$ is based on sub-sampling scheme S2. Further, let the Assumptions 1, 2 and 3 hold, almost surely. Then, for some absolute constants $c, C > 0$, with probability at least $1 - e^{-p}$, the updates of the form stated in Eq. (1.2) satisfy*

$$\|\hat{\theta}^{t+1} - \theta_*\|_2 \leq \xi_1^t \|\hat{\theta}^t - \theta_*\|_2 + \xi_2^t \|\hat{\theta}^t - \theta_*\|_2^2,$$

for coefficients ξ_1^t, ξ_2^t defined as

$$\xi_1^t = \left\| I - \eta_t \mathbf{Q}^t \mathbf{H}_{S_t}(\hat{\theta}^t) \right\|_2 + \eta_t \left\| \mathbf{Q}^t \right\|_2 \times cK \sqrt{\frac{p}{|S_t|} \log \left(\frac{\operatorname{diam}(\mathcal{C})^2 \left(M_n + M_{|S_t|} \right)^2 |S_t|}{K^2} \right)},$$

$$\xi_2^t = \eta_t \frac{M_n}{2} \left\| \mathbf{Q}^t \right\|_2.$$

Proof of Lemma A.1. The first part of the proof is the same as Lemma 3.1. We carry our analysis from Eq.(A.1). Note that in this general set-up, the iterates are random variables that depend on the random functions. Therefore, we use a uniform bound for the right hand side in Eq.(A.1). That is,

$$\begin{aligned} \left\| I - \eta_t \mathbf{Q}^t \widetilde{\mathbf{H}} \right\|_2 \leq & \left\| I - \eta_t \mathbf{Q}^t \mathbf{H}_S(\hat{\theta}^t) \right\|_2 \\ & + \eta_t \left\| \mathbf{Q}^t \right\|_2 \Big\{ \sup_{\theta \in \mathcal{C}} \left\| \mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta) \right\|_2 + \frac{M_n}{2} \|\hat{\theta}^t - \theta_*\|_2 \Big\}. \end{aligned}$$

By the Assumption 1, given $\theta, \theta' \in \mathcal{C}$ such that $\|\theta - \theta'\|_2 \leq \Delta$, we have,

$$\begin{aligned} \left\| \mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta) \right\|_2 &\leq \left\| \mathbf{H}_S(\theta') - \mathbf{H}_{[n]}(\theta') \right\|_2 + \left(M_n + M_{|S|} \right) \|\theta - \theta'\|_2 \\ &\leq \left\| \mathbf{H}_S(\theta') - \mathbf{H}_{[n]}(\theta') \right\|_2 + \left(M_n + M_{|S|} \right) \Delta. \end{aligned}$$

Next, we will use a covering net argument to obtain a bound on the matrix empirical process. Note that similar bounds on the matrix forms can be obtained through other approaches like *chaining* as well [DE15]. Let $\mathcal{T}_\Delta$ be a Δ-net over the convex set $\mathcal{C}$. By the above inequality, we obtain

$$\sup_{\theta \in \mathcal{C}} \left\| \mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta) \right\|_2 \leq \max_{\theta' \in \mathcal{T}_\Delta} \left\| \mathbf{H}_S(\theta') - \mathbf{H}_{[n]}(\theta') \right\|_2 + \left(M_n + M_{|S|} \right) \Delta. \tag{A.5}$$

Now we will argue that the right hand side is small with high probability using the matrix Hoeffding's inequality from [Tro12]. By the union bound over $\mathcal{T}_\Delta$, we have

$$\mathbb{P}\left(\max_{\theta' \in \mathcal{T}_\Delta} \left\| \mathbf{H}_S(\theta') - \mathbf{H}_{[n]}(\theta') \right\|_2 > \epsilon \right) \leq |\mathcal{T}_\Delta| \; \mathbb{P}\left(\left\| \mathbf{H}_S(\theta') - \mathbf{H}_{[n]}(\theta') \right\|_2 > \epsilon \right).$$

For the first term on the right hand side, by Lemma E.1, we write:

$$|\mathcal{T}_\Delta| \leq \left(\frac{\operatorname{diam}(\mathcal{C})}{2\Delta/\sqrt{p}} \right)^p.$$

As before, let $S = \{s_1, s_2, ..., s_{|S|}\}$, that is, s_i denote the different indices in S. For any $\theta \in \mathcal{C}$ and $i = 1, 2, ..., n$, we define the centered Hessians $\mathbf{W}_i(\theta)$ as

$$\mathbf{W}_i(\theta) = \mathbf{H}_{s_i}(\theta) - \mathbf{H}_{[n]}(\theta).$$

By the Assumption (2), we have the same bounds as in Eq.(A.2). Hence, for $\epsilon > 0$ and $\theta \in \mathcal{C}$, by the matrix Hoeffding's inequality [Tro12],

$$\mathbb{P}\left(\left\|\mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta)\right\|_2 > \epsilon\right) \leq \ 2p \exp\left\{-\frac{|S|\epsilon^2}{32K^2}\right\}.$$

We would like to obtain an exponential decay with a rate of at least $\mathcal{O}(p)$. Hence, we require,

$$\begin{aligned} p\log\left(\frac{\mathrm{diam}(\mathcal{C})\sqrt{p}}{2\Delta}\right) + \log(2p) + p \leq \ & p\log\left(\frac{4\mathrm{diam}(\mathcal{C})\sqrt{p}}{\Delta}\right), \\ \leq & \frac{|S|\epsilon^2}{32K^2}, \end{aligned}$$

which gives the optimal value of ϵ as

$$\epsilon \geq \sqrt{\frac{32K^2p}{|S|}\log\left(\frac{4\mathrm{diam}(\mathcal{C})\sqrt{p}}{\Delta}\right)}.$$

Therefore, we conclude that for the above choice of ϵ, with probability at least $1 - e^{-p}$, we have

$$\max_{\theta \in \mathcal{T}_\Delta} \left\|\mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta)\right\|_2 < \sqrt{\frac{32K^2p}{|S|}\log\left(\frac{4\mathrm{diam}(\mathcal{C})\sqrt{p}}{\Delta}\right)}.$$

Applying this result to the inequality in Eq.(A.5), we obtain that with probability at least $1 - e^{-p}$,

$$\sup_{\theta \in \mathcal{C}} \left\|\mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta)\right\|_2 \leq \sqrt{\frac{32K^2p}{|S|}\log\left(\frac{4\mathrm{diam}(\mathcal{C})\sqrt{p}}{\Delta}\right)} + \left(M_n + M_{|S|}\right)\Delta.$$

The right hand side of the above inequality depends on the net covering diameter Δ. We optimize over Δ using Lemma E.5 which provides for

$$\Delta = 4\sqrt{\frac{K^2p}{\left(M_n + M_{|S|}\right)^2|S|}\log\left(\frac{\mathrm{diam}(\mathcal{C})^2\left(M_n + M_{|S|}\right)^2|S|}{K^2}\right)},$$

we obtain that with probability at least $1 - e^{-p}$,

$$\sup_{\theta \in \mathcal{C}} \left\|\mathbf{H}_S(\theta) - \mathbf{H}_{[n]}(\theta)\right\|_2 \leq 8K\sqrt{\frac{p}{|S|}\log\left(\frac{\mathrm{diam}(\mathcal{C})^2\left(M_n + M_{|S|}\right)^2|S|}{K^2}\right)}.$$

Combining this with the bound stated in Eq.(A.1), we conclude the proof. □

Proof of Theorem 3.6.

$$\begin{aligned}\left|\xi_1^t - \xi_1^*\right| =& \left|\frac{\lambda_p^t}{\lambda_{r+1}^t} - \frac{\lambda_p^*}{\lambda_{r+1}^*}\right| + cK\sqrt{\frac{\log(p)}{|S_t|}}\left|\frac{1}{\lambda_{r+1}^t} - \frac{1}{\lambda_{r+1}^*}\right| \\ \leq& \frac{K|\lambda_{r+1}^t - \lambda_{r+1}^*| + K|\lambda_p^t - \lambda_p^*|}{\lambda_{r+1}^*\lambda_{r+1}^t} + cK\sqrt{\frac{\log(p)}{|S_t|}}\frac{|\lambda_{r+1}^t - \lambda_{r+1}^*|}{\lambda_{r+1}^*\lambda_{r+1}^t}\end{aligned}$$

By the Weyl's and matrix Hoeffding's [Tro12] inequalities (See Eq. (A.3) for details), we can write

$$|\lambda_j^t - \lambda_j^*| \leq \left\|\mathbf{H}_{S_t}(\hat{\theta}^t) - \mathbf{H}_{[n]}(\theta_*)\right\|_2 \leq cK\sqrt{\frac{\log(p)}{|S_t|}},$$

with probability $1 - 2/p$. Then,

$$\begin{aligned}\left|\xi_1^t - \xi_1^*\right| \leq& \frac{c'K\sqrt{\frac{\log(p)}{|S_t|}}}{\lambda_{r+1}^*\lambda_{r+1}^t} + \frac{c''K^2\frac{\log(p)}{|S_t|}}{\lambda_{r+1}^*\lambda_{r+1}^t}, \\ \leq& \frac{c'''K\sqrt{\frac{\log(p)}{|S_t|}}}{k\left(k - cK\sqrt{\frac{\log(p)}{|S_t|}}\right)},\end{aligned}$$

for some constants c and c'''. □

Proof of Corollary 4.1. Observe that $f_i(\theta) = \Phi(\langle x_i, \theta\rangle) - y_i\langle x_i, \theta\rangle$, and $\boldsymbol{\nabla}_\theta^2 f_i(\theta) = x_i x_i^T \Phi^{(2)}(\langle x_i, \theta\rangle)$. For an index set S, we have $\forall \theta, \theta' \in \mathcal{C}$

$$\begin{aligned}\left\|\mathbf{H}_S(\theta) - \mathbf{H}_S(\theta')\right\|_2 =& \left\|\frac{1}{|S|}\sum_{i\in S} x_i x_i^T\left[\Phi^{(2)}(\langle x_i, \theta\rangle) - \Phi^{(2)}(\langle x_i, \theta'\rangle)\right]\right\|_2, \\ \leq& L\max_{i\in S}\|x_i\|_2^3\ \|\theta - \theta'\|_2 \leq LR_x^{3/2}\ \|\theta - \theta'\|_2.\end{aligned}$$

Therefore, the Assumption 1 is satisfied with the Lipschitz constant $M_{|S_t|} \coloneqq LR_x^{3/2}$. Moreover, by the inequality

$$\left\|\boldsymbol{\nabla}_\theta^2 f_i(\theta)\right\|_2 = \|x_i\|_2^2\ \Phi^{(2)}(\langle x_i, \theta\rangle) \leq R_x, = \left\|x_i x_i^T \Phi^{(2)}(\langle x_i, \theta\rangle)\right\|_2$$

the Assumption 2 is satisfied for $K \coloneqq R_x$. We conclude the proof by applying Theorem 3.2. □

B Properties of composite convergence

In the previous sections, we showed that NewSamp gets a composite convergence rate, i.e., the ℓ_2 distance from the current iterate to the optimal value can be bounded by the sum of a linearly and a quadratically converging term. We study such convergence rates assuming the coefficients do not change at each iteration t. Denote by Δ_t, the aforementioned ℓ_2 distance at iteration step t, i.e.,

$$\Delta_t = \|\hat{\theta}^t - \theta_*\|_2, \tag{B.1}$$

and assume that the algorithm gets a composite convergence rate as

$$\forall t \geq 0, \qquad \Delta_{t+1} \leq \xi_1 \Delta_t + \xi_2 \Delta_t^2,$$

where $\xi_1, \xi_2 > 0$ denote the coefficients of linearly and quadratically converging terms, respectively.

B.1 Local asymptotic rate

We state the following theorem on the local convergence properties of compositely converging algorithms.

Lemma B.1. *For a compositely converging algorithm as in Eq. (B.1) with coefficients* $1 > \xi_1, \xi_2 > 0$*, if the initial distance* Δ_0 *satisfies* $\Delta_0 < (1 - \xi_1)/\xi_2$*, then we have*

$$\limsup_{t \to \infty} -\frac{1}{t} \log(\Delta_t) \leq -\log(\xi_1).$$

The above theorem states that the local convergence of a compositely converging algorithm will be dominated by the linear term.

Proof of Lemma B.1. The condition on the initial point implies that $\Delta_t \to 0$ as $t \to \infty$. Hence, for any given $\delta > 0$, there exists a positive integer T such that $\forall t \geq T$, we have $\Delta_t < \delta/\xi_2$. For such values of t, we write

$$\xi_1 + \xi_2 \Delta_t < \xi_1 + \delta,$$

and using this inequality we obtain

$$\Delta_{t+1} < (\xi_1 + \delta)\Delta_t.$$

The convergence of above recursion gives

$$-\frac{1}{t} \log(\Delta_t) < -\log(\xi_1 + \delta) - \frac{1}{t} \log(\Delta_0).$$

Taking the limit on both sides concludes the proof. □

B.2 Number of iterations

The total number of iterations, combined with the per-iteration cost, determines the total complexity of an algorithm. Therefore, it is important to derive an upper bound on the total number of iterations of a compositely converging algorithm.

Lemma B.2. *For a compositely converging algorithm as in Eq. (B.1) with coefficients* $\xi_1, \xi_2 \in (0, 1)$*, assume that the initial distance* Δ_0 *satisfies* $\Delta_0 < (1 - \xi_1)/\xi_2$ *and for a given tolerance* ϵ*, define the interval*

$$D = \left(\max \left\{ \epsilon, \frac{\xi_1 \Delta_0}{1 - \xi_2 \Delta_0} \right\}, \Delta_0 \right).$$

Then the total number of iterations needed to approximate the true minimizer with ϵ *tolerance is upper bounded by* $T(\delta_*)$*, where*

$$\delta_* = \operatorname{argmin}_{\delta \in D} T(\delta)$$

and

$$T(\delta) = \log_2 \left(\frac{\log\left(\xi_1 + \delta\xi_2\right)}{\log\left(\frac{\Delta_0}{\delta}(\xi_1 + \delta\xi_2)\right)} \right) + \frac{\log\left(\frac{\epsilon}{\delta}\right)}{\log(\xi_1 + \xi_2\delta)}.$$

Proof of Lemma B.2. We have $\Delta_t \to 0$ as $t \to \infty$ by the condition on initial point Δ_0. Let $\delta \in D$ be a real number and t_1 be the last iteration step such that $\Delta_t > \delta$. Then $\forall t \geq t_1$,

$$\begin{aligned} \Delta_{t+1} \leq & \xi_1 \Delta_t + \xi_2 \Delta_t^2, \\ \leq & \left(\frac{\xi_1}{\delta} + \xi_2 \right) \Delta_t^2. \end{aligned}$$

Therefore, in this regime, the convergence rate of the algorithm is dominated by a quadratically converging term with coefficient $(\xi_1/\delta + \xi_2)$. The total number of iterations needed to attain a tolerance of δ is upper bounded by

$$t_1 \leq \log_2 \left(\frac{\log\left(\xi_1 + \delta\xi_2\right)}{\log\left(\frac{\Delta_0}{\delta}(\xi_1 + \delta\xi_2)\right)} \right).$$

When $\Delta_t < \delta$, namely $t > t_1$, we have

$$\begin{aligned} \Delta_{t+1} \leq & \xi_1 \Delta_t + \xi_2 \Delta_t^2, \\ \leq & \left(\xi_1 + \xi_2\delta\right) \Delta_t. \end{aligned}$$

In this regime, the convergence rate is dominated by a linearly converging term with coefficient $(\xi_1 + \xi_2\delta)$. Therefore, the total number of iterations since t_1 until a tolerance of ϵ is reached can be upper bounded by

$$t_2 \leq \frac{\log\left(\frac{\epsilon}{\delta}\right)}{\log(\xi_1 + \xi_2\delta)}.$$

Hence, the total number of iterations needed for a composite algorithm as in Eq. B.1 to reach a tolerance of ϵ is upper bounded by

$$T(\delta) = t_1 + t_2 = \log_2 \left(\frac{\log\left(\xi_1 + \delta\xi_2\right)}{\log\left(\frac{\Delta_0}{\delta}(\xi_1 + \delta\xi_2)\right)} \right) + \frac{\log\left(\frac{\epsilon}{\delta}\right)}{\log(\xi_1 + \xi_2\delta)}.$$

The above statement holds for any $\delta \in D$. Therefore, we minimize $T(\delta)$ over the set D. □

C Choosing the step size η_t

In most optimization algorithms, step size plays a crucial role. If the dataset is so large that one cannot try out many values of the step size. In this section, we describe an efficient and adaptive way for this purpose by using the theoretical results derived in the previous sections.

In the proof of Lemma 3.1, we observe that the convergence rate of NewSamp is governed by the term

$$\left\|\mathbf{I}-\eta_t\mathbf{Q}^tH_{[n]}(\tilde{\theta})\right\|_2 \leq \left\|\mathbf{I}-\eta_t\mathbf{Q}^tH_{[n]}(\hat{\theta}^t)\right\|_2 + \eta_t\left\|\mathbf{Q}^t\right\|_2\left\|H_{[n]}(\hat{\theta}^t)-H_{[n]}(\tilde{\theta})\right\|_2$$

where $\mathbf{Q}^t$ is defined as in Algorithm 1. The right hand side of the above equality has a linear dependence on η_t. We will see later that this term has no effect in choosing the right step size. On the other hand, the first term on the right hand size can be written as,

$$\left\|\mathbf{I}-\eta_t\mathbf{Q}^tH_{[n]}(\hat{\theta}^t)\right\|_2 = \max\left\{1-\eta_t\lambda_{\min}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t)), \eta_t\lambda_{\max}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t))-1\right\}.$$

If we optimize the above quantity over η_t, we obtain the optimal step size as

$$\eta_t = \frac{2}{\lambda_{\min}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t)) + \lambda_{\max}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t))}. \tag{C.1}$$

It is worth mentioning that for the Newton's method where $\mathbf{Q}^t = H_{[n]}(\hat{\theta}^t)^{-1}$, the above quantity is equal to 1.

Since NewSamp does not compute the full Hessian $\mathbf{H}_{[n]}(\hat{\theta}^t)$ (which would take $\mathcal{O}(np^2)$ computation), we will relate the quantity in Eq. (C.1) to the first few eigenvalues of $\mathbf{Q}^t$. Therefore, our goal is to relate the eigenvalues of $\mathbf{Q}^tH_{[n]}(\hat{\theta}^t)$ to that of $\mathbf{Q}^t$.

By the Lipschitz continuity of eigenvalues , we write

$$\begin{aligned}\left|1-\lambda_{\max}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t))\right| \leq& \left\|\mathbf{Q}^t\right\|_2\left\|H_S(\hat{\theta}^t)-H_{[n]}(\hat{\theta}^t)\right\|_2, \\ =& \frac{1}{\lambda^t_{r+1}}\mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right).\end{aligned} \tag{C.2}$$

Similarly, for the minimum eigenvalue, we can write

$$\left|\frac{\lambda^t_p}{\lambda^t_{r+1}} - \lambda_{\min}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t))\right| \leq \frac{1}{\lambda_{r+1}}\mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right). \tag{C.3}$$

One might be temped to use 1 and $\lambda^t_p/\lambda^t_{r+1}$ for the minimum and the maximum eigenvalues of $\mathbf{Q}^tH_{[n]}(\hat{\theta}^t)$, but the optimal values might be slightly different from these values if the sample size is chosen to be small. On the other hand, the eigenvalues λ^t_{r+1} and λ^t_p can be computed with $\mathcal{O}(p^2)$ cost and we already know the order of the error term. That is, one can calculate λ^t_{r+1} and λ^t_p and use the error bounds to correct the estimate.

The eigenvalues of the sample covariance matrix will concentrate around the true values, spreading to be larger for large eigenvalues and smaller for the small eigenvalues. That is, if we will we will overestimate if we estimate λ_1 with λ^t_1. Therefore, if we use 1, we will always underestimate the value of $\lambda_{\max}(\mathbf{Q}^tH_{[n]}(\hat{\theta}^t))$, which, based on Eq. (C.2) and Eq. (C.3), suggests a correction term of $\mathcal{O}\left(\sqrt{\log(p)/|S|}\right)$. Further, the top $r+1$ eigenvalues of $[Q^t]^{-1}$ are close to the eigenvalues of $H_{[n]}(\hat{\theta}^t)$, but shifted upwards if $p/2 > r$. When $p/2 < r$, we see an opposite behavior. Hence, we add or subtract a correction term of order $\mathcal{O}\left(\sqrt{\log(p)/|S|}\right)$ to $\lambda^t_p/\lambda^t_{r+1}$ whether $p/2 > r$ or $p/2 < r$, respectively. The corrected estimators could be written as

$$\widehat{\lambda_{\max}}\left(\mathbf{Q}^t H_{[n]}(\hat{\theta}^t)\right) = 1 + \mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right),$$

$$\widehat{\lambda_{\min}}\left(\mathbf{Q}^t H_{[n]}(\hat{\theta}^t)\right) = \frac{\lambda_p}{\lambda_{r+1}} + \mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right) \qquad \text{if } p/2 > r,$$

$$= \frac{\lambda_p}{\lambda_{r+1}} - \mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right) \qquad \text{if } p/2 < r.$$

We are more interested in the case where $p/2 > r$. In this case, we suggest the step size for the iteration step t as

$$\eta_t = \frac{2}{1 + \frac{\lambda_p^t}{\lambda_{r+1}^t} + \mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right)}$$

which uses the eigenvalues that are already computed to construct $\mathbf{Q}^t$. Contrary to the most algorithms, the optimal step size of NewSamp is generally larger than 1.

D Further experiments and details

In this section, we present the details of the experiments presented in Figure 2 and provide additional simulation results.

We first start with additional experiments. The goal of this experiment is to further analyze the effect of rank in the performance of NewSamp . We experimented using r-spiked model for $r = 3, 10, 20$. The case $r = 3$ was already presented in Figure 2, which is included in Figure 3 to ease the comparison. The results are presented in Figures 3 and the details are summarized in Table 2. In the case of LR optimization, we observe through Figure 3 that stochastic algorithms enjoy fast convergence in the beginning but slows down later as they get close to the true minimizer. The algorithms that come closer to NewSamp in terms of performance are BFGS and LBFGS. Especially when $r = 20$, performance of BFGS and that of NewSamp are similar, yet NewSamp still does better. In the case of SVM optimization, the algorithm that comes closer to NewSamp is Newton's method.

We further demonstrate how the algorithm coefficients ξ_1 and ξ_2 between datasets in Figure 4.

E Useful lemmas

Lemma E.1. *Let $\mathcal{C}$ be convex and bounded set in $\mathbb{R}^p$ and T_ϵ be an ϵ-net over $\mathcal{C}$. Then,*

$$|T_\epsilon| \leq \left(\frac{\text{diam}(\mathcal{C})}{2\epsilon/\sqrt{p}}\right)^p.$$

Proof of Lemma E.1. A similar proof appears in [VdVW96]. The set $\mathcal{C}$ can be contained in a p-dimensional cube of size $\text{diam}(\mathcal{C})$. Consider a grid over this cube with mesh width $2\epsilon/\sqrt{p}$. Then $\mathcal{C}$ can be covered with at most $(\text{diam}(\mathcal{C})/(2\epsilon/\sqrt{p}))^p$ many cubes of edge length $2\epsilon/\sqrt{p}$. If ones takes

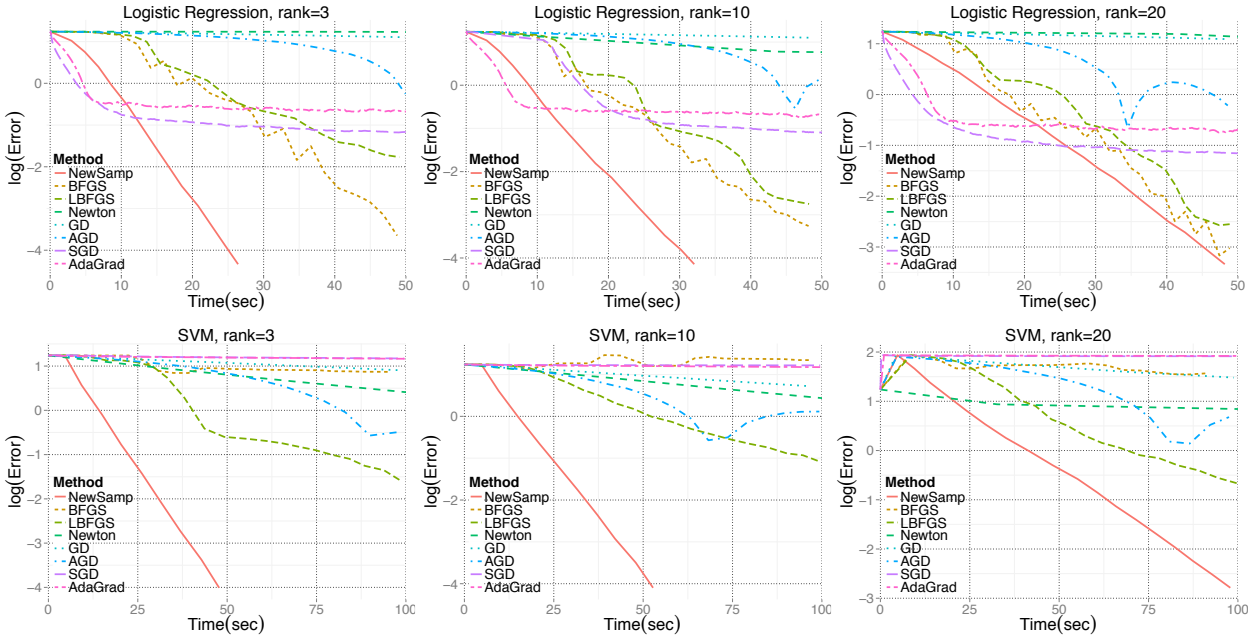

Figure 3: The plots demonstrate the behavior of several optimization methods on a synthetic data set for training SVMs. The elapsed time in seconds versus log of ℓ_2-distance to the true minimizer is plotted. Red color represents the proposed method NewSamp .

the projection of the centers of such cubes onto $\mathcal{C}$ and considers the circumscribed balls of radius ϵ, we may conclude that $\mathcal{C}$ can be covered with at most

$$\left(\frac{\operatorname{diam}(\mathcal{C})}{2\epsilon/\sqrt{p}}\right)^p$$

many balls of radius ϵ. □

Lemma E.2 ([Ver10]). *Let X be a symmetric $p \times p$ matrix, and let T_ϵ be an ϵ-net over S^{p-1}. Then,*

$$\|X\|_2 \leq \frac{1}{1-2\epsilon} \sup_{v \in T_\epsilon} |\langle Xv, v\rangle| .$$

Lemma E.3 ([GN10]). *Let $\mathcal{X}$ be a finite set of Hermitian matrices in $\mathbb{R}^{p\times p}$ where $\forall X_i \in \mathcal{X}$, we have*

$$\mathbb{E}[X_i] = 0, \qquad \|X_i\|_2 \leq \gamma, \qquad \left\|\mathbb{E}[X_i^2]\right\|_2 \leq \sigma^2.$$

Given its size, let S denote a uniformly random sample from $\{1, 2, ..., |\mathcal{X}|\}$ with or without replacement. Then we have

$$\mathbb{P}\left(\left\|\frac{1}{|S|}\sum_{i\in S} X_i\right\|_2 > \epsilon\right) \leq 2p \exp\left\{-|S| \min\left(\frac{\epsilon^2}{4\sigma^2}, \frac{\epsilon}{2\gamma}\right)\right\}.$$

Logistic Regression

	Rank=3		Rank=10		Rank=20	
Method	Elapsed(sec)	Iter	Elapsed(sec)	Iter	Elapsed(sec)	Iter
NewSamp	26.412	12	32.059	15	55.995	26
BFGS	50.699	22	54.756	31	56.606	34
LBFGS	103.590	47	64.617	37	107.708	67
Newton	18235.842	449	35533.516	941	31032.893	777
GD	345.025	198	322.671	198	311.946	197
AGD	449.724	233	436.282	272	450.734	290

Support Vector Machines

	Rank=3		Rank=10		Rank=20	
Method	Elapsed(sec)	Iter	Elapsed(sec)	Iter	Elapsed(sec)	Iter
NewSamp	47.755	8	52.767	9	124.989	22
BFGS	13352.254	2439	10672.657	2219	21874.637	4290
LBFGS	326.526	67	218.706	44	275.991	55
Newton	775.191	16	734.480	16	4159.486	106
GD	1512.305	238	1089.413	237	1518.063	269
AGD	1695.44	239	1066.484	238	1874.75	294

Table 2: Details of the simulations presented in Figures 3.

Lemma E.4. *Let Z be a random variable with a density function f and cumulative distribution function F. If $F^C = 1 - F$, then,*

$$\left|\mathbb{E}[Z\mathbb{1}_{\{|Z|>t\}}]\right| \leq t\mathbb{P}(|Z| > t) + \int_t^\infty \mathbb{P}(|Z| > z)dz.$$

Proof. We write,

$$\mathbb{E}[Z\mathbb{1}_{\{|Z|>t\}}] = \int_t^\infty zf(z)dz + \int_{-\infty}^{-t} zf(z)dz.$$

Using integration by parts, we obtain

$$\begin{aligned}\int zf(z)dz =& -zF^C(z) + \int F^C(z)dz,\\ =& zF(z) - \int F(z)dz.\end{aligned}$$

Since $\lim_{z\to\infty} zF^C(z) = \lim_{z\to-\infty} zF(z) = 0$, we have

$$\begin{aligned}\int_t^\infty zf(z)dz =& tF^C(t) + \int_t^\infty F^C(z)dz,\\ \int_{-\infty}^{-t} zf(z)dz =& -tF(-t) - \int_{-\infty}^{-t} F(z)dz,\\ =& -tF(-t) - \int_t^\infty F(-z)dz.\end{aligned}$$

CT Slices Dataset

	LR		SVM	
Method	Elapsed(sec)	Iter	Elapsed(sec)	Iter
NewSamp	9.488	19	22.228	33
BFGS	9.568	38	2094.330	5668
LBFGS	51.919	217	165.261	467
Newton	14.162	5	58.562	25
GD	350.863	2317	1660.190	4828
AGD	176.302	915	1221.392	3635

MSD Dataset

	LR		SVM	
Method	Elapsed(sec)	Iter	Elapsed(sec)	Iter
NewSamp	25.770	38	71.755	49
BFGS	43.537	75	9063.971	6317
LBFGS	81.835	143	429.957	301
Newton	144.121	30	100.375	18
GD	642.523	1129	2875.719	1847
AGD	397.912	701	1327.913	876

Synthetic Dataset

	LR		SVM	
Method	Elapsed(sec)	Iter	Elapsed(sec)	Iter
NewSamp	26.412	12	47.755	8
BFGS	50.699	22	13352.254	2439
LBFGS	103.590	47	326.526	67
Newton	18235.842	449	775.191	16
GD	345.025	198	1512.305	238
AGD	449.724	233	1695.44	239

Table 3: Details of the experiments presented in Figure 2.

Hence, we obtain the following bound,

$$\begin{aligned}\left|\mathbb{E}[Z\mathbb{1}_{\{|Z|>t\}}]\right| =& \left|tF^C(t) + \int_t^\infty F^C(z)dz - tF(-t) - \int_t^\infty F(-z)dz\right|, \\ \leq& t\left(F^C(t) + F(-t)\right) + \left(\int_t^\infty F^C(z) + F(-z)dz\right), \\ \leq& t\mathbb{P}(|Z| > t) + \int_t^\infty \mathbb{P}(|Z| > z)dz.\end{aligned}$$

□

Lemma E.5. *For $a, b > 0$, and ϵ satisfying*

$$\epsilon = \left\{\frac{a}{2}\log\left(\frac{2b^2}{a}\right)\right\}^{1/2} \qquad \text{and} \qquad \frac{2}{a}b^2 > e,$$

we have $\epsilon^2 \geq a\log(b/\epsilon)$.

Proof. Since $a, b > 0$ and $x \to e^x$ is a monotone increasing function, the above inequality condition is equivalent to

$$\frac{2\epsilon^2}{a}e^{\frac{2\epsilon^2}{a}} \geq \frac{2b^2}{a}.$$

Now, we define the function $f(w) = we^w$ for $w > 0$. f is continuous and invertible on $[0, \infty)$. Note that f^{-1} is also a continuous and increasing function for $w > 0$. Therefore, we have

$$\epsilon^2 \geq \frac{a}{2}f^{-1}\left(\frac{2b^2}{a}\right)$$

Observe that the smallest possible value for ϵ would be simply the square root of $af^{-1}\left(2b^2/a\right)/2$. For simplicity, we will obtain a more interpretable expression for ϵ. By the definition of f^{-1}, we have

$$\log(f^{-1}(y)) + f^{-1}(y) = \log(y).$$

Since the condition on a and b enforces $f^{-1}(y)$ to be larger than 1, we obtain the simple inequality that

$$f^{-1}(y) \leq \log(y).$$

Using the above inequality, if ϵ satisfies

$$\epsilon^2 = \frac{a}{2}\log\left(\frac{2b^2}{a}\right) \geq \frac{a}{2}g\left(\frac{2b^2}{a}\right),$$

we obtain the desired inequality. □

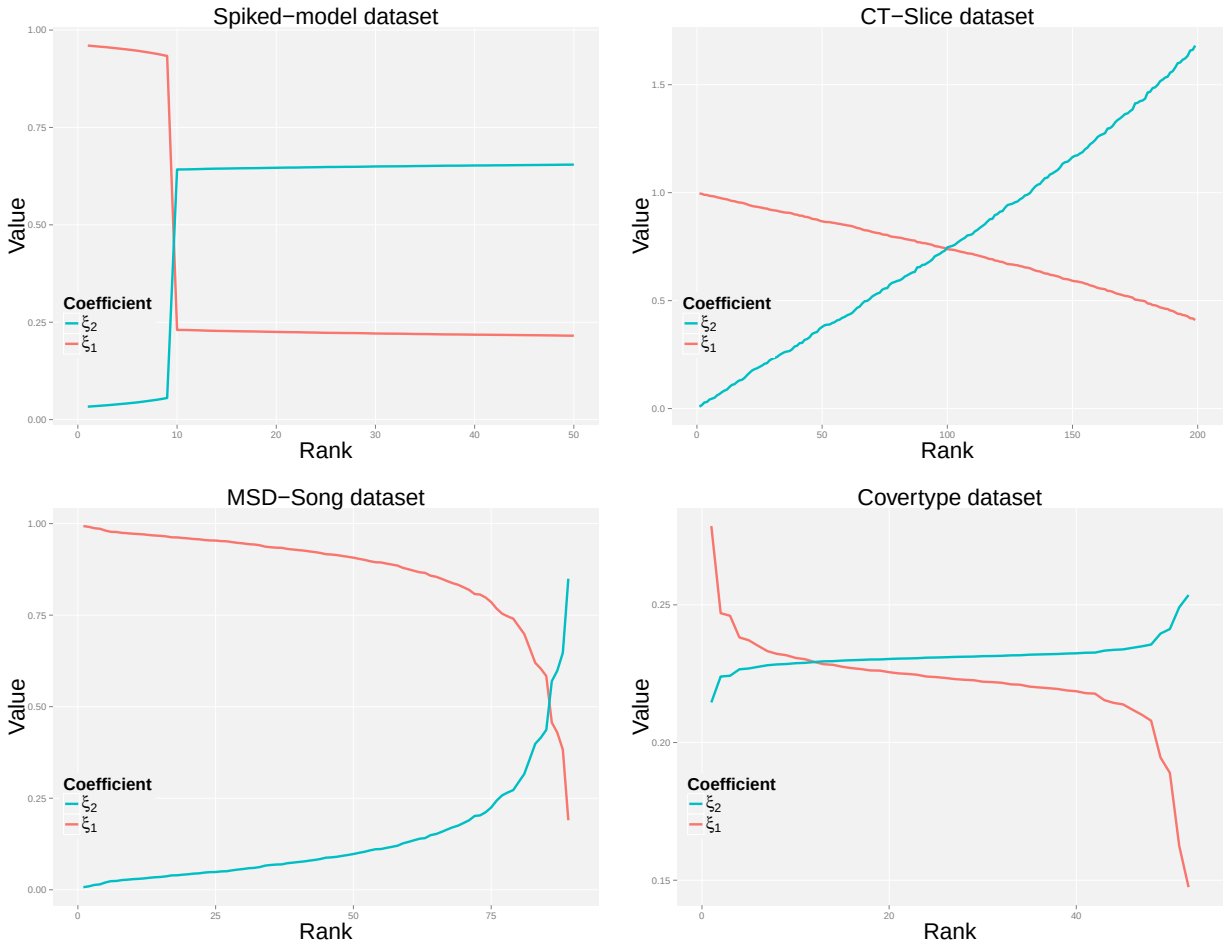

Figure 4: The plots demonstrate the behavior of ξ_1 and ξ_2 over several datasets.