Convergence rates of sub-sampled Newton methods

Authored by:

Sham M. Kakade
Sujay Sanghavi
Kamalika Chaudhuri
Praneeth Netrapalli

Abstract

An active learner is given a class of models, a large set of unlabeled examples, and the ability to interactively query labels of a subset of these examples; the goal of the learner is to learn a model in the class that fits the data well. Previous theoretical work has rigorously characterized label complexity of active learning, but most of this work has focused on the PAC or the agnostic PAC model. In this paper, we shift our attention to a more general setting – maximum likelihood estimation. Provided certain conditions hold on the model class, we provide a two-stage active learning algorithm for this problem. The conditions we require are fairly general, and cover the widely popular class of Generalized Linear Models, which in turn, include models for binary and multi-class classification, regression, and conditional random fields. We provide an upper bound on the label requirement of our algorithm, and a lower bound that matches it up to lower order terms. Our analysis shows that unlike binary classification in the realizable case, just a single extraround of interaction is sufficient to achieve near-optimal performance in maximum likelihood estimation. On the empirical side, the recent work in (Gu et al. 2012) and (Gu et al. 2014) (on active linear and logistic regression) shows the promise of this approach.

1 Paper Body

We consider the problem of minimizing a sum of n functions via projected iterations onto a convex parameter set C ? Rp , where n p 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 low-rank approximation 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 focus on the following minimization problem, n

minimize f (?) :=

1X fi (?), n i=1

(1.1)

where fi : Rp ! R. Most machine learning models can be expressed as above, where each function fi 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 [Bis95, BV04, Nes04]. For a given convex set C ? Rp , we denote the Euclidean projection onto this set by PC . We consider the updates of the form ? ? ??t+1 = PC ??t ?t Qt r? f (??t) , (1.2)

where ?t is the step size and Qt is a suitable scaling matrix that provides curvature information. Updates of the form Eq. (1.2) have been extensively studied in the optimization literature (for simplicity, we assume C = Rp throughout the introduction). The case where Qt is equal to identity matrix corresponds to Gradient Descent (GD) which, under smoothness assumptions, achieves linear convergence rate with O(np) per-iteration cost. More precisely, GD with ideal step size yields t ?t ?? k2 , where, as limt!1 ? t = 1 (? / ?), and ? is the i-th largest k??t+1 ?? k2 ? ?1, p GD k? 1,GD 1 i eigenvalue of the Hessian of f (?) at minimizer ?? . Second order methods such as Newton?s Method (NM) and Natural Gradient Descent (NGD) [Ama98] can be recovered by taking Qt to be the inverse Hessian and the Fisher information evaluated at the current iterate, respectively. Such methods may achieve quadratic convergence rates with 1

O(np2 + p3) per-iteration cost [Bis95, Nes04]. In particular, for t large enough, Newton?s method yields k??t+1 ?? k2 ? ?2,NM k??t ?? k22 , and it is insensitive to the condition number of the Hessian. However, when the number of samples grows large, computing Qt becomes extremely expensive. A popular line of research tries to construct the matrix Qt 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 utilized, resulting in an efficient update on Qt . A celebrated Quasi-Newton method is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm which requires O(np + p2) per-iteration cost [Bis95, Nes04]. An alternative approach is to use sub-sampling techniques, where scaling matrix Qt is based on randomly selected set of data points [Mar10, BCNN11, VP12, Erd15]. 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 (?)), but is a poor approximation in the directions corresponding to small eigenvalues (flatter directions in f (?)). 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 k??t+1 ?? k2 ? ?1t k??t ?? k2 + ?2t k??t ?? k22 with coefficient that are explicitly given (and are computable from data). 2. The asymptiotic behavior of the linear convergence coefficient is limt!1 ?1t = 1 (?p / ?r+1) + , for small. The condition number (?1 / ?p) which controls the convergence of GD, has been replaced by the milder (?r+1 / ?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 ?t = 1. 4. The complexity per iteration of NewSamp is O(np + —S—p2) with —S— the sample size. 5. Our theoretical results can be used to obtain convergence rates of previously proposed subsampling algorithms. 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 n and p are large but not so large as to make gradient computations (of order np) and matrix manipulations (of order p3) prohibitive. 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. Improvements to SGD have been developed by incorporating the previous gradient directions in the current update equation [SRB13, 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,

Erd15]. If the sub-sampled Hessian is close to the true Hessian, these methods can approach NM in terms of convergence rate, nevertheless, they enjoy 2

Algorithm 1 NewSamp Input: ??0 , r, ?, {?t }t , t = 0. 1. Define: PC (?) = argmin?0 2C k? ?0 k2 is the Euclidean projection onto C, [Uk , ?k] = TruncatedSVDk (H) is rank-k truncated SVD of H with ?ii = i . 2. while k??t+1 ??t k2 ? ? do Sub-sample a set of indices St ? [n]. P Let HSt = —S1t — i2St r2? fi (??t), and [Ur+1 , ?r+1] = TruncatedSVDr+1 (HSt), 1 1 T Qt = r+1 Ip + U r ? r 1 r+1 ? ? Ir U r , ??t+1 = PC ??t ?t Qt r? f (??t) , t t + 1. 3. end while Output: ??t .

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]). On the further improvements of the sub-sampling algorithms, a common approach is to use Conjugate Gradient (CG) methods and/or Krylov sub-spaces [Mar10, BCNN11, VP12]. Lastly, there are various hybrid algorithms that combine two or more techniques to increase the performance. Examples include, sub-sampling and Quasi-Newton [BHNS14], SGD and GD [FS12], NGD and NM [LRF10], NGD and low-rank approximation [LRMB08].

2

NewSamp : Newton-Sampling method via rank thresholding

In the regime we consider, n p, there are two main drawbacks associated with the classical second order methods such as Newton?s method. The dominant issue is the computation of the Hessian matrix, which requires O(np2) operations, and the other issue is inverting the Hessian, which requires O(p3) computation. Sub-sampling is an effective and efficient way of tackling the first issue. 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]. 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]. To this extent, we use low-rank approximation as the important second order information is generally contained in the largest few eigenvalues/vectors of the Hessian. 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 St , —St — and HSt , respectively. Assuming that the functions fi ?s are convex, eigenvalues of the symmetric matrix HSt are non-negative. Therefore, SVD and eigenvalue decomposition coincide. The operation TruncatedSVDk (HSt) = [Uk , ?k] is the best rank-k approximation, i.e., takes HSt as input and returns the largest k eigenvalues ?k 2 Rk?k with the corresponding k eigenvectors Uk 2 Rp?k . This procedure requires O(kp2) computation [HMT11]. Operator PC projects the current iterate to the feasible set C using Euclidean projection. We assume that

this projection can be done efficiently. To construct the curvature matrix [Qt] 1 , instead of using the basic rank-r approximation, we fill its 0 eigenvalues with the (r + 1)-th eigenvalue of the sub-sampled Hessian which is the largest eigenvalue below the threshold. If we compute a truncated SVD with k = r + 1 and ?ii = i , the described operation results in 1 1 T Qt = r+1 Ip + U r ? r 1 (2.1) r+1 Ir Ur , 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 O np + (—St — + r)p2 ? O np + —St —p2 . NewSamp takes the parameters {?t , —St —}t and r as inputs. We discuss in Section 3.4, how to choose them optimally, based on the theory in Section 3. 3

Convergence Rate

Convergence Coefficients

0

0.25

?2

Value

log(Error)

?1

0.20

?3

?4

?5

Sub?sample size NewSamp : St = 100 NewSamp : St = 200 NewSamp : St = 500 0

200

Iterations

Coefficient ?1 : linear ?2 : quadratic

0.15 400

0

600

20

Rank

40

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 up to a scaling constant. x-axis shows the number of kept eigenvalues. Plots are obtained using Covertype dataset.

By the construction of Qt , 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. We will explain this phenomenon in Section 3, by Theorems 3.2 and 3.3. The right plot in Figure 1 demonstrates how the coefficients of two phases depend on the thresholded rank. 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 subsampling schemes: S1: Independent sub-sampling: At each iteration t, St is uniformly sampled from [n] = {1, 2, ..., n}, independently from the sets {S? }? ¡t , with or without replacement. S2: Sequentially dependent sub-sampling: At each iteration t, St is sampled from [n], based on a distribution which might depend on the previous sets {S? }? ¡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 subsampling scheme S2. In our theoretical analysis, we make the following assumptions: Assumption 1 (Lipschitz continuity). For any subset S ? [n], 9M—S— depending on the size of S, such that 8?, ?0 2 C, kHS (?) HS (?0)k2 ? M—S— k? ?0 k2 . Assumption 2 (Bounded Hessian). 8i 2 [n], r2? fi (?) is upper bounded by a constant K, i.e., max r2? fi (?) i?n

2

? K.

3.1 Independent sub-sampling In this section, we assume that St ? [n] is sampled according to the sub-sampling scheme S1. In fact, many stochastic algorithms assume that St is a uniform subset of [n], because in this case the sub-sampled Hessian is an unbiased estimator of the full Hessian. That is, 8? 2 C, E [HSt (?)] = H[n] (?), where the expectation is over the randomness in St . We next show that for any scaling matrix Qt that is formed by the sub-samples St , iterations of the form Eq. (1.2) will have a composite convergence rate, i.e., combination of a linear and a quadratic phases. 4

Lemma 3.1. Assume that the parameter set C is convex and St ? [n] is based on sub-sampling scheme S1 and sufficiently large. Further, let the Assumptions 1 and 2 hold and ?? 2 C. Then, for an absolute constant c ¿ 0, with probability at least 1 2/p, the updates of the form Eq. (1.2) satisfy k??t+1 ?? k2 ? ? t k??t ?? k2 + ? t k??t ?? k2 , 1

2

for coefficients ?1t and ?2t defined as ?1t

= I

?t Q HSt (??t) t

2

+ ?t cK Q
t 2
s
2
log(p) , —St —
?2t = ?t
Mn Qt 2
2
.

Remark 1. If the initial point ??0 is close to ?? , 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 Qt . In particular, if we choose Qt = HSt1 , we obtain a bound for the simple sub-sampled Hessian method. In this case, the coefficients ?1t and ?2t depend on kQt k2 = 1/ tp where tp is the smallest eigenvalue of the sub-sampled Hessian. Note that tp 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 ti , the i-th eigenvalue of HSt (??t) where ??t is given by NewSamp at iteration step t. If the step size satisfies 2 ?t ? , (3.1) 1 + tp / tr+1 then we have, with probability at least 1 2/p, k??t+1 ?? k2 ? ? t k??t

?? k2 + ?2t k??t
1
?? k22 ,

for an absolute constant c ¿ 0, for the coefficients ?1t and ?2t are defined as s t cK log(p) Mn p ?1t = 1 ?t t + ?t t , ?2t = ?t t . —St — 2 r+1 r+1 r+1 NewSamp has a composite convergence rate where ?1t and ?2t 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 ?t = 1 is feasible for the conditions of Theorem 3.2. 3.2 Sequentially dependent sub-sampling Here, we assume that the sub-sampling scheme S2 is used to generate {S? }? 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 z1 , z2 , ..., zn 2 Z be i.i.d. observations from a distribution D. For a fixed ? 2 Rp and 8i 2 [n], we assume that the functions {fi }ni=1 satisfy fi (?) = '(zi , ?), for some function ' : Z ? Rp ! R. Most statistical learning algorithms can be formulated as above, e.g., in classification problems, one has access to i.i.d. samples {(yi , xi)}ni=1 where yi and xi denote the class label and the covariate, and ' measures the classification error (See Section 4 for examples). For sub-sampling scheme S2, an analogue of Lemma 3.1 is stated in Appendix as Lemma B.1, which leads to the following result. Theorem 3.3. Assume that the parameter set C is convex and St ? [n] is based on the sub-sampling scheme S2. Further, let the Assumptions 1, 2 and

3 hold, almost surely. Conditioned on the event E = {?? 2 C}, if the step size satisfies Eq. 3.1, then for ??t given by NewSamp at iteration t, with probability at least 1 cE e p for cE = c/P(E), we have k??t+1 ?? k2 ? ? t k??t ?? k2 + ? t k??t ?? k2 , 1

2

for the coefficients ?1t and ?2t defined as s ? 2 t 0 diam(C) Mn + M—St — c K p p ?1t = 1 ?t t + ?t t log —St — K2 r+1 r+1 where c, c0 ¿ 0 are absolute constants and

t i

2

2

—St —

?

,

?2t = ?t

denotes the i-th eigenvalue of HSt (??t). 5

Mn 2

, t r+1

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 C.

3.3 Dependence of coefficients on t and convergence guarantees The coefficients ?1t and ?2t depend on the iteration step which is an undesirable aspect of the above results. However, these constants can be well approximated by their analogues ?1? and ?2? evaluated at the optimum which are defined by simply replacing tj with ?j in their definition, where the latter is the j-th eigenvalue of full-Hessian at ?? . For the sake of simplicity, we only consider the case where the functions ? ! fi (?) are quadratic. Theorem 3.4. Assume that the functions fi (?) are quadratic, St is based on scheme S1 and ?t = 1. Let the full Hessian at ?? be lower bounded by k. Then for sufficiently large —St — and absolute constants c1 , c2 , with probability 1 2/p p c1 K log(p)/—St — t ? := . p ?1 ?1 ? k k c2 K log(p)/—St —

Theorem 3.4 implies that, when the sub-sampling size is sufficiently large, ?1t will concentrate around ?1? . Generalizing the above theorem to non-quadratic functions is straightforward, in which case, one would get additional terms involving the difference k??t ?? k2 . In the case of scheme S2, if one uses fixed sub-samples, then the coefficient ?1t 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 [Erd15, EM15]. Corollary 3.5. Assume that ?1t and ?2t are well-approximated by ?1? and ?2? with an error bound of , i.e., ?it ? ?i? + for i = 1, 2, as in Theorem 3.4. For the initial point ??0 , a sufficient condition for convergence is 1 ?1? k??0 ?? k2 ¡ . ?2? + 3.4 Choosing the algorithm parameters Step size: Let = O(log(p)/—St —). We suggest the following step size for NewSamp at iteration t, 2 ?t ( ) = . (3.2) t 1 + p /

tr+1 + Note that ?t (0) is the upper bound in Theorems 3.2 and 3.3 and it minimizes the first component of ?1t . The other terms in ?1t and ?2t linearly depend on ?t . To compensate for that, we shrink ?t (0) towards 1. Contrary to most algorithms, optimal step size of NewSamp is larger than 1. A rigorous derivation of Eq. 3.2 can be found in [EM15]. Sample size: By Theorem 3.2, a sub-sample of size O((K/ ?p)2 log(p)) should be sufficient to obtain a small coefficient for the linear phase. Also note that sub-sample size —St — 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 O(R log(p)) samples [Ver10]. 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 4.1

Examples Generalized Linear Models (GLM)

Maximum likelihood estimation in a GLM setting is equivalent to minimizing the negative loglikelihood ‘(?), n 1X minimize f (?) = [(hxi , ?i) yi hxi , ?i] , (4.1) ?2C n i=1

where is the cumulant generating function, xi 2 Rp denote the rows of design matrix X 2 Rn?p , and ? 2 Rp is the coefficient vector. Here, hx, ?i denotes the inner product between the vectors x, ?. The function defines the type of GLM, i.e., (z) = z 2 gives ordinary least squares (OLS) and (z) = log(1 + ez) gives logistic regression (LR). Using the results from Section 3, we perform a convergence analysis of our algorithm on a GLM problem. 6

Corollary 4.1. Let St ? [n] be a uniform sub-sample, and C = Rp be the parameter set. Assume that the second derivative of the cumulant generating function, (2) is bounded by 1, and it is Lipschitz continuous p with Lipschitz constant L.pFurther, assume that the covariates are contained in a ball of radius Rx , i.e. maxi2[n] kxi k2 ? Rx . Then, for ??t given by NewSamp with constant step size ?t = 1 at iteration t, with probability at least 1 2/p, we have k??t+1

for constants ?1t and ?2t defined as

?? k2 ? ?1t k??t

t r+1

s

where c ¿ 0 is an absolute constant and

t i

?1t

4.2

=1

t i t r+1

+

cRx

?? k2 + ?2t k??t

log(p) , —St —

?? k22 , 3/2

?2t =

LRx , 2 tr+1

is the ith eigenvalue of HSt (??t).

Support Vector Machines (SVM)

A linear 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 [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 O(n3) complexity) the primal might be better-suited for optimization of linear SVMs [Cha07]. The primal problem for the linear SVM can be written as n 1 1 X minimize f (?) = k?k22 + C ‘(yi , h?, xi i) (4.2) ?2C 2 2 i=1 where (yi , xi) denote the data samples, ? defines the separating hyperplane, C ¿ 0 and ‘ 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, or 2 Hinge-2 loss [Cha07]. In the case of Hinge-2 loss, i.e., ‘(y, h?, xi) = max {0, 1 yh?, xi} , by combining the offset and the normal vector of the hyperplane into a single parameter vector ?, and denoting by SVt the set of indices of all the support vectors at iteration t, we may write the Hessian, o X 1 n r2? f (?) = I+C xi xTi , where SVt = {i : yi h?t , xi i ¡ 1}. —SVt — i2SVt

When —SVt — 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 numerical studies. We experimented on two optimization problems, namely, Logistic Regression (LR) and SVM. LR minimizes Eq. 4.1 for the logistic function, whereas SVM minimizes Eq. 4.2 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. 3. Newton?s Method (NM) achieves a quadratic convergence rate by utilizing the inverse Hessian evaluated at the current iterate. 4. Broyden-Fletcher-Goldfarb-Shanno (BFGS) is the most popular and stable Quasi-Newton method. Qt is formed by accumulating the information from iterates and gradients. 5. Limited Memory BFGS (L-BFGS) is a variant of BFGS, which uses only the recent iterates and gradients to construct Qt , providing improvement in terms of memory usage. 6. Stochastic Gradient Descent (SGD) is a simplified version of GD where, at each iteration, a randomly selected gradient is used. We follow the guidelines of [Bot10] for the step size.

7

Dataset:)
Synthe'c) Logistic Regression, rank=3
1
MSD) Logistic Regression, rank=60
1
log(Error)
Method NewSamp BFGS LBFGS Newton GD AGD ?4 SGD AdaGrad ?2
0
10
20
30
40
Time(sec)
50
0
log(Error)
0
0
log(Error)
CT)Slices) Logistic Regression, rank=60
?1
Method NewSamp BFGS LBFGS Newton ?3 GD AGD SGD AdaGrad ?4 0
?2
SVM, rank=3
5
Time(sec)
10
15
?1
Method NewSamp BFGS LBFGS Newton ?3 GD AGD SGD AdaGrad ?4 0
10 ?2
SVM, rank=60
20
30
40
Time(sec)
50
SVM, rank=60
1 2
2
?1
Method NewSamp ?2 BFGS LBFGS Newton ?3 GD AGD SGD ?4 AdaGrad
0
25
50
Time(sec)

75
100
log(Error)
log(Error)
log(Error)
0 0
Method NewSamp BFGS LBFGS ?2 Newton GD AGD SGD AdaGrad ?4 0
10
Time(sec)
20
30
0
Method NewSamp BFGS LBFGS ?2 Newton GD AGD SGD AdaGrad ?4 0
30
60
Time(sec)
90
120

Figure 2: Performance of several algorithms on different datasets. NewSamp is represented with red color .

7. Adaptive Gradient Scaling (AdaGrad) uses an adaptive learning rate based on the previous gradients. AdaGrad significantly improves the performance and stability of SGD [DHS11]. For batch algorithms, we used constant step size and for all the algorithms, the step size that provides the fastest convergence is chosen. For stochastic algorithms, we optimized over the parameters that define the step size. Parameters of NewSamp are selected following the guidelines in Section 3.4. We experimented over various datasets that are given in Table 1. Each dataset consists of a design matrix X 2 Rn?p and the corresponding observations (classes) y 2 Rn . Synthetic data is generated through a multivariate Gaussian distribution. 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 —St — 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 methods enjoy fast convergence at start, but slows down after several epochs. The algorithm that comes close to NewSamp in terms of performance is BFGS. In the case of SVM, NM is the closest algorithm to NewSamp . Note that the global P convergence of BFGS is not better than that of GD [Nes04]. The condition for super-linear rate is t k?t ?? k2 ¡ 1 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 other second order methods. For NewSamp, even though rank thresholding provides a level of robustness, we found that initial point is still an important factor. Details about Figure 2 and additional experiments can be found in Appendix C. Dataset CT slices Covertype MSD Synthetic

n 53500 581012 515345 500000
p 386 54 90 300
r 60 20 60 3
Reference [GKS+ 11, Lic13] [BD99, Lic13] [MEWL, Lic13] ?
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 p and has O np + —S—p2 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 demonstrate the performance of the proposed algorithm which we compared to the classical optimization methods. 8

2 References

[Ama98] Shun-Ichi Amari, Natural gradient works efficiently in learning, Neural computation 10 (1998). [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 (2011). [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, Compag (1999). [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, 1995. [Bot10] L'eon Bottou, Large-scale machine learning with stochastic gradient descent, COMPSTAT, 2010. [BV04] Stephen Boyd and Lieven Vandenberghe, Convex optimization, Cambridge University Press, 2004. [CCS10] Jian-Feng Cai, Emmanuel J Cand'es, 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 (2007). [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, J. Mach. Learn. Res. 12 (2011), 2121?2159. [DM77] John E Dennis, Jr and Jorge J Mor?e, Quasi-newton methods, motivation and theory, SIAM review 19 (1977), 46?89. [EM15] Murat A Erdogdu and Andrea Montanari, Conver-

gence rates of sub-sampled Newton methods, arXiv preprint arXiv:1508.02810 (2015). [Erd15] Murat A. Erdogdu, Newton-Stein Method: A second order method for GLMs via Stein?s lemma, NIPS, 2015. [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. [GKS+ 11] Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Sebastian P?olsterl, and Alexander Cavallaro, 2d image registration in ct images using radial image descriptors, MICCAI 2011, Springer, 2011. [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. [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. [Lic13] M. Lichman, UCI machine learning repository, 2013. [LRF10] Nicolas Le Roux and Andrew W Fitzgibbon, A fast natural newton method, ICML, 2010. [LRMB08] Nicolas Le Roux, Pierre-A Manzagol, and Yoshua Bengio, Topmoumoute online natural gradient algorithm, NIPS, 2008. [Mar10] James Martens, Deep learning via hessian-free optimization, ICML, 2010, pp. 735?742. [MEWL] Thierry B. Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere, The million song dataset, ISMIR-11. [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. [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 Sch?olkopf and Alexander J Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, MIT press, 2002. [Tro12] Joel A Tropp, User-friendly tail bounds for sums of random matrices, Foundations of Computational Mathematics (2012). [Ver10] Roman Vershynin, Introduction to the non-asymptotic analysis of random matrices, arXiv:1011.3027 (2010). [VP12] Oriol Vinyals and Daniel Povey, Krylov Subspace Descent for Deep Learning, AISTATS, 2012.

9