OPT2020: 12th Annual Workshop on Optimization for Machine Learning

SGB: Stochastic Gradient Bound Method for Optimizing Partition Functions

Jing Wang JW5665@NYU.EDU
Anna Choromanska AC5455@NYU.EDU
Department of Electrical and Computer Engineering, New York University

Abstract

This paper addresses the problem of optimizing partition functions in a stochastic learning setting. We propose a stochastic variant of the bound majorization algorithm from [29] that relies on upper-bounding the partition function with a quadratic surrogate. The update of the proposed method, that we refer to as Stochastic Partition Function Bound (SPFB), resembles scaled stochastic gradient descent where the scaling factor relies on a second order term that is however different from the Hessian. Similarly to quasi-Newton schemes, this term is constructed using the stochastic approximation of the value of the function and its gradient. We prove sub-linear convergence rate of the proposed method and show the construction of its low-rank variant (LSPFB). Experiments on logistic regression demonstrate that the proposed schemes significantly outperform SGD. We also discuss how to use quadratic partition function bound for efficient training of deep learning models and in non-convex optimization.

1. Introduction

The problem of estimating the probability density function over a set of random variables underlies majority of learning frameworks and heavily depends on the partition function. Partition function is a normalizer of a density function and ensures that it integrates to 1. This function needs to be minimized when learning proper data distribution. Optimizing the partition function however is a hard and often intractable problem [20]. It has been addressed in a number of ways in the literature. Below we review strategies that directly confront the partition function (we skip pseudo-likelihood strategies [24] and score matching [25] and ratio matching [26] techniques, which avoid direct partition function computations).

There exists a variety of Markov chain Monte Carlo methods for approximately maximizing the likelihood of models with partition functions such as i) contrastive divergence [13, 22] and persistent contrastive divergence [42], which perform Gibbs sampling and are used inside a gradient descent procedure to compute model parameter update, and ii) fast persistent contrastive divergence [43], which relies on re-parameterizing the model and introducing the parameters that are trained with much larger learning rate such that the Markov chain is forced to mix rapidly. The above mentioned techniques are Gibbs sampling strategies that focus on estimating the gradient of the log-partition function. Another technique called noise-contrastive estimation [21] treats partition function like an additional model parameter whose estimate can be learned via nonlinear logistic regression discriminating between the observed data and some artificially generated noise. Methods that directly estimate the partiton function rely on importance sampling. More conretely they estimate the ratio of the partition functions of two models, where one of the partition function is known. The extensions of this technique, annealed importance sampling [27, 37] and bridge sampling [6], cope with the setting where two considered distributions are far from each other.

Finally, bound majorization constitutes yet another strategy for performing density estimation. Bound majorization methods iteratively construct and optimize variational bound on the original optimization problem. Among these techniques we have iterative scaling schemes [8, 17], EM algorithm [2, 18], non-negative matrix factorization method [34], convex-concave procedure [47], minimization by incremental surrogate optimization [35], and technique based on constructing quadratic partition function bound [29] (early predecessors of these techniques include [9, 33]). The latter technique uses tighter bound compared to the aforementioned methods, and exhibits faster convergence compared to generic first- [36, 40, 44] and second-order [1, 7, 48] techniques in the batch optimization setting for both convex and non-convex learning problems. In this paper we revisit the quadratic bound majorization technique and propose its stochastic variant that we analyze both theoretically and empirically. We prove its convergence rate and show that it is performing favorably compared to SGD [10, 38]. Finally, we propose future research directions that can utilize quadratic partition function bound in non-convex optimization, including deep learning setting. With a pressing need to develop landscape-driven deep learning optimization strategies [3–5, 14, 14, 15, 19, 23, 28, 30, 31, 39, 41, 45], we foresee the resurgence of interest in bound majorization techniques and its applicability to non-convex learning problems.

2. Method

Consider the log-linear model given by a density function of the form

$$\mathrm{p}(y|\mathbf{x},\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}}(y))/Z_{\mathbf{x}}(\boldsymbol{\theta}), \tag{2.1}$$

where $(\mathbf{x}, y)$ is the observation-label pair ($y \in \{1, 2, ..., n\}$), $\mathbf{f}_{\mathbf{x}} : \{1, 2, ..., n\} \rightarrow \mathbb{R}^d$ represents a feature map, $\boldsymbol{\theta} \in \mathbb{R}^d$ is a model parameter vector, and $Z_{\mathbf{x}}(\boldsymbol{\theta}) = \sum_{y=1}^{n} \exp(\boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}}(y))$ is the partition function. Maximum likelihood framework estimates $\boldsymbol{\theta}$ from a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^T$ by maximizing the objective function of the form

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{T} \log \mathrm{p}(y_i|\mathbf{x}_i,\boldsymbol{\theta}) - \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 = \sum_{i=1}^{T}\left[\boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}_i}(y) - \log Z_{\mathbf{x}_i}(\boldsymbol{\theta})\right] - \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \tag{2.2}$$

where the second term is a regularization (λ is a regularization coefficient). This framework and its various extensions underlie logistic regression, conditional random fields, maximum entropy estimation, latent likelihood, deep belief networks, and other density estimation approaches. Equation 2.2 requires minimizing the partition function $Z_{\mathbf{x}}(\boldsymbol{\theta})$. This can be done by optimizing the variational quadratic bound on the partition function instead. The bound is shown in Theorem 1.

Algorithm 1: Partition function bound

Input: $\widetilde{\boldsymbol{\theta}} \in \mathbb{R}^d$, observation $\mathbf{x}$, $\mathbf{f}_{\mathbf{x}}(y) \forall y \in \{1, ..., n\}$.
Output: Bound parameters: $\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$, z

Init $z \rightarrow 0^+$, $\boldsymbol{\mu} = \mathbf{0}$, $\boldsymbol{\Sigma} = z\mathbf{I}$
for $y = 1, ..., n$ **do**
  $\alpha_j = \exp(\widetilde{\boldsymbol{\theta}}^T\mathbf{f}_{\mathbf{x}}(y)); \mathbf{l} = \mathbf{f}_{\mathbf{x}}(y) - \boldsymbol{\mu}$
  $\beta = \frac{\tanh(\frac{1}{2}\log(\alpha/z))}{2\log(\alpha/z)}; \kappa = \frac{\alpha}{z+\alpha}$
  $\boldsymbol{\Sigma} += \beta\mathbf{l}\mathbf{l}^T$
  $\boldsymbol{\mu} += \kappa\mathbf{l}$
  $z += \alpha$
end

Algorithm 2: Maximum Likelihood via Stochastic Partition Function Bound (SPFB)

Input: initial parameters $\boldsymbol{\theta}_0$, training data set $\{(\mathbf{x}_j, y_j)\}_{j=1}^T$, features $\mathbf{f}_{\mathbf{x}_j}$, learning rates η_t, regularization coefficient λ
Output: Model parameters $\boldsymbol{\theta}$

Set $\boldsymbol{\theta} = \boldsymbol{\theta}_0$
while *not converged* **do**
  randomly select a training point $(\mathbf{x}_t, y_t)$
  Get $\boldsymbol{\Sigma}_t$, $\boldsymbol{\mu}_t$ from $\mathbf{f}_{\mathbf{x}_t}$, $\boldsymbol{\theta}$ via Algorithm 1
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_t\left(\boldsymbol{\Sigma}_t + \lambda\mathbf{I}\right)^{-1}\left(\boldsymbol{\mu}_t - \mathbf{f}_{\mathbf{x}_t}(y_t) + \lambda\boldsymbol{\theta}\right)$
end

Theorem 1. [29] *Let* $Z_{\mathbf{x}}(\boldsymbol{\theta}) = \sum_{y=1}^{n} \exp\left(\boldsymbol{\theta}^T \mathbf{f}_{\mathbf{x}}(y)\right)$. *Algorithm 1 finds* $z, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ *such that*

$$Z_{\mathbf{x}}(\boldsymbol{\theta}) \leq z \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \boldsymbol{\Sigma} (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \boldsymbol{\mu}\right) \tag{2.3}$$

for any $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}, \mathbf{f}_{\mathbf{x}}(y) \in \mathbb{R}^d$ *for any* $y \in \{1, ..., n\}$.

2.1. Stochastic Partition Function Bound (SPFB)

The partition function bound of Algorithm 1 can be used to optimize the objective in Equation 2.2. The maximum likelihood parameter update given by the bound takes the form:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \left(\sum_{j=1}^{T} \boldsymbol{\Sigma}_j + \lambda \mathbf{I}\right)^{-1} \left(\sum_{j=1}^{T} \left[\boldsymbol{\mu}_j - \mathbf{f}_{\mathbf{x}_j}(y_j)\right] + \lambda \boldsymbol{\theta}^t\right), \tag{2.4}$$

where $\boldsymbol{\Sigma}_j$s and $\boldsymbol{\mu}_j$s are computed from Algorithm 1. In contrast to the above full-batch update, Stochastic Partition Function Bound (SPFB) method that we propose in Algorithm 2 updates parameters after seeing each training data point, rather than the entire data set, according to the formula:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t \left(\boldsymbol{\Sigma}_t + \lambda \mathbf{I}\right)^{-1} \left(\boldsymbol{\mu}_t - \mathbf{f}_{\mathbf{x}_t}(y_t) + \lambda \boldsymbol{\theta}^t\right), \tag{2.5}$$

where $\eta_t = \eta_0/t$ is the learning rate. Denote $f(\boldsymbol{\theta}; \mathbf{x}_t) = \log(Z_t(\boldsymbol{\theta})) - \boldsymbol{\theta}^T \mathbf{f}_{\mathbf{x}_t}(\mathbf{y}_t) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$ to be an unbiased estimation of objective function $L(\boldsymbol{\theta})$, where $L(\boldsymbol{\theta}) = -J(\boldsymbol{\theta})$. The above formula (2.5) can be rewritten as

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t \left(\boldsymbol{\Sigma}_t + \lambda \mathbf{I}\right)^{-1} \nabla f(\boldsymbol{\theta}^t; \mathbf{x}_t). \tag{2.6}$$

The next theorem shows the convergence rate of SPFB.

Theorem 2. $\{\boldsymbol{\theta}^t\}$ *is the sequence of parameters generated by Algorithm 2. There exists* $0 < \mu_1 < \mu_2$, $0 < \lambda_1 < \lambda_2$ *such that for all iterations* t,

$$\mu_1 \mathbf{I} \prec \left(\boldsymbol{\Sigma}_t + \lambda \mathbf{I}\right)^{-1} \prec \mu_2 \mathbf{I} \quad \textit{and} \quad \lambda_1 \mathbf{I} \prec \nabla^2 L(\boldsymbol{\theta}) \prec \lambda_2 \mathbf{I}, \tag{2.7}$$

and there exists a constant σ, *such that for all* $\boldsymbol{\theta} \in \mathbb{R}^d$, $\mathbf{E}_{\mathbf{x}_t}[\|\nabla f(\boldsymbol{\theta}; \mathbf{x}_t)\|]^2 \leq \sigma^2$. *Define the learning rate in iteration* t *as* $\eta_t = \eta_0/t$, *where* $\eta_0 > 1/(2\mu_1\lambda_1)$. *Then for all* $t > 1$,

$$\mathbf{E}[L(\boldsymbol{\theta}^t) - L(\boldsymbol{\theta}^*)] \leq Q(\eta_0)/t, \tag{2.8}$$

where $Q(\eta_0) = \max\left\{\frac{\lambda_2 \mu_2^2 \eta_0^2 \sigma^2}{2(2\mu_1\lambda_1\eta_0 - 1)}, L(\boldsymbol{\theta}^1) - L(\boldsymbol{\theta}^*)\right\}$.

Theorem 2 guarantees sub-linear convergence rate for SPFB when the step size is diminishing. However, the time complexity of SPFB is $O(nd^2 + d^3) = \widetilde{O}(d^3)$, due to the computation and inversion of matrix $\boldsymbol{\Sigma}_t$, which is less appealing than the $O(nd)$ complexity of SGD. This is next addressed.

2.2. Low-rank bound

In this section, we provide a low-rank construction of the bound that applies to both batch and stochastic setting. We decompose matrix $\boldsymbol{\Sigma}$ into $\boldsymbol{\Sigma} = \mathbf{V}^T \mathbf{S} \mathbf{V} + \mathbf{D}$, where $\mathbf{V} \in \mathbb{R}^{k \times d}$ (orthnormal matrix), $\mathbf{S} \in \mathbb{R}^{k \times k}$, and $\mathbf{D} \in \mathbb{R}^{d \times d}$ (diagonal matrix) and apply Woodbury formula to compute the inverse: $\boldsymbol{\Sigma}^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{V}^T(\mathbf{S}^{-1} + \mathbf{V}\mathbf{D}^{-1}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}^{-1}$ (clearly, the inverse only requires $O(k^3)$ time and does not affect the total time complexity when rank $k \ll d$). Note that Algorithm 1 performs rank-one update to matrix $\boldsymbol{\Sigma}$ of the form: $\boldsymbol{\Sigma} = \boldsymbol{\Sigma} + \mathbf{r}\mathbf{r}^T$, where $\mathbf{r} = \sqrt{\beta}\mathbf{l}$. This update can be "projected" onto matrices $\mathbf{V}, \mathbf{S}$, and $\mathbf{D}$. The concrete updates of matrices $\mathbf{V}, \mathbf{S}$, and $\mathbf{D}$ are shown in Algorithm 3. The next theorem, Theorem 3, guarantees that the low-rank bound is indeed an upper-bound on the partition function [1].

1. We simultaneously repair the low-rank bound construction of [29], which breaks this property.

Theorem 3. *Let* $Z_{\mathbf{x}}(\boldsymbol{\theta}) = \sum_{y=1}^{n} \exp\left(\boldsymbol{\theta}^T \mathbf{f}_{\mathbf{x}}(y)\right)$. *In each iteration of the* x*-loop in Algorithm 3 finds* $z, \boldsymbol{\mu}, \mathbf{V}, \mathbf{S}, \mathbf{D}$ *such that*

$$Z_{\mathbf{x}}(\boldsymbol{\theta}) \leq z \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T(\mathbf{V}^T\mathbf{S}\mathbf{V} + \mathbf{D})(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T\boldsymbol{\mu}\right) \tag{2.9}$$

for any $\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}, \mathbf{f}_{\mathbf{x}}(y) \in \mathbb{R}^d$ *for any* $y \in \{1, ..., n\}$.

Low-rank variant of Algorithm 2 is presented in Algorithm 4. Note that all proofs supporting this section are deferred to the Supplement.

Algorithm 3: Low-rank Partition Function Bound

Input: $\widetilde{\boldsymbol{\theta}} \in \mathbb{R}^d$, observation $\mathbf{x}$, $\mathbf{f}_{\mathbf{x}}(y) \forall y \in \{1, ..., n\}$, rank $k \in \mathbb{N}$
Output: Low-rank bound parameters: $\mathbf{V}$, $\mathbf{S}$, $\mathbf{D}$, $\boldsymbol{\mu}$, z

$z \to 0^+$, $\mathbf{S} = 0$, $\mathbf{V} = orthonormal \in \mathbb{R}^{k \times d}$, $\mathbf{D} = z\mathbf{I}$, $\boldsymbol{\mu} = \mathbf{0}$
for *each sample* $\mathbf{x}_j$ *in batch* **do** // x-loop
  Init $z_j \leftarrow 0^+$, $\boldsymbol{\upsilon} = 0$
  for *each label* $y \in \{1, 2, ..., n\}$ **do**
    $\alpha = \exp(\widetilde{\boldsymbol{\theta}}^T \mathbf{f}_{\mathbf{x}_j}(y))$; $\mathbf{r} = \sqrt{\frac{\tanh(\frac{1}{2}\log(\alpha/z))}{2\log(\alpha/z)}}(\mathbf{f}_{\mathbf{x}_j}(y) - \boldsymbol{\upsilon})$;
    $\mathbf{p} = \mathbf{V}\mathbf{r}$; $\mathbf{a} = \mathbf{V}^T\mathbf{p}$; $\mathbf{g} = \mathbf{r} - \mathbf{a}$, $\mathbf{S} += \mathbf{p}\mathbf{p}^T$
    $\mathbf{Q}^T\mathbf{A}\mathbf{Q} = svd(\mathbf{S})$; $\mathbf{S} \leftarrow \mathbf{A}$; $\mathbf{V} \leftarrow \mathbf{Q}\mathbf{V}$; $\mathbf{D} += \|\mathbf{g}\|\|\mathbf{a}\|\mathbf{I} \in \mathbb{R}^{d \times d}$
    $\mathbf{s} = [\mathbf{S}(1,1), ..., \mathbf{S}(k,k), \|\mathbf{g}\|^2]^T$, $\widetilde{k} = \arg\min_{i=1,...,k+1} \mathbf{s}(i)$
    if $\widetilde{k} \leq k$ **then**
      $\mathbf{D} = \mathbf{D} + \mathbf{S}(\widetilde{k}, \widetilde{k}) 1^T |\mathbf{V}(\widetilde{k}, \cdot)| diag(|\mathbf{V}(\widetilde{k}, \cdot)|)$
      $\mathbf{S}(\widetilde{k}, \widetilde{k}) = \|\mathbf{g}\|^2$; $\mathbf{g} = \frac{\mathbf{g}}{\|\mathbf{g}\|}$; $\mathbf{V}(\widetilde{k}, \cdot) = \mathbf{g}$
    else
      $\mathbf{D} += 1^T |\mathbf{g}| diag(|\mathbf{g}|)$
    $\boldsymbol{\upsilon} += \frac{\alpha}{z_j + \alpha}(\mathbf{f}_{\mathbf{x}_j}(y) - \boldsymbol{\upsilon})$; $z_j += \alpha$
  end
  $\boldsymbol{\mu} += \boldsymbol{\upsilon}$, $z += z_j$
end

Algorithm 4: MLE via Low-rank Stochastic Partition Function Bound (LSPFB)

Input: initial parameters $\boldsymbol{\theta}_0$, training data set $\{(\mathbf{x}_j, y_j)\}_{j=1}^T$, features $\mathbf{f}_{\mathbf{x}_j}$, learning rates η_t, regularization coefficient λ
Output: Model parameters $\boldsymbol{\theta}$

Set $\boldsymbol{\theta} = \boldsymbol{\theta}_0$
while *not converged* **do**
  randomly select a training point $(\mathbf{x}_t, y_t)$
  Get $\mathbf{V}_t$, $\mathbf{S}_t$, $\mathbf{D}_t$, $\boldsymbol{\mu}_t$ from $\mathbf{x}_t$, $\mathbf{f}_{\mathbf{x}_t}$, $\boldsymbol{\theta}$ via Algorithm 3 (input batch is a single data point $\mathbf{x}_t$)
  $\mathbf{D}_t = \mathbf{D}_t + \lambda\mathbf{I}$; $\boldsymbol{\mu}_t = \boldsymbol{\mu}_t - \mathbf{f}_{\mathbf{x}_t}(y_t) + \lambda\boldsymbol{\theta}$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_t \left(\mathbf{D}_t^{-1} - \mathbf{D}_t^{-1}\mathbf{V}_t^T \left(\mathbf{S}_t^{-1} + \mathbf{V}_t\mathbf{D}_t^{-1}\mathbf{V}_t^T\right)^{-1} \mathbf{V}_t\mathbf{D}_t^{-1}\right) \boldsymbol{\mu}_t$
end

3. Experiments

Experiments were performed on adult[1] ($T = 48842$, $n = 2$, $d = 14$), KMNIST[2] ($T = 60000$, $n = 10$, $d = 784$), and Fashion-MNIST[2] ($T = 60000$, $n = 10$, $d = 784$) data sets. All algorithms were run with mini-batch size equal to $m = 1000$. For low-rank methods, we explored the following settings of the rank k: $k = 1, 5, 10, 100$. Hyperparameters for all methods were chosen to achieve the best test performance. We compare SPFB and LSPFB with SGD for ℓ_2-regularized logistic regression on the adult, KMNIST, and Fashion-MNIST data sets. Both SPFB and LSPFB show clear advantage over SGD in terms of convergence speed.

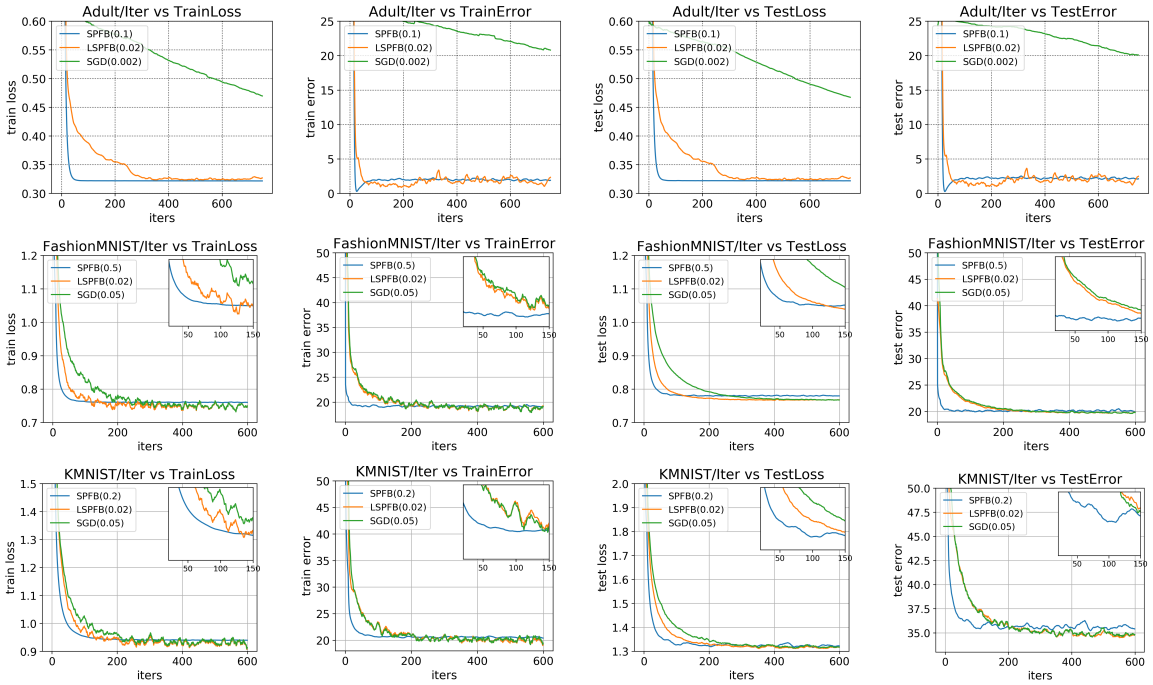

Figure 1: A Comparison of SPFB, LSPFB, and SGD on $l2$-regularized logistic regression problem.

4. Discussion

Here we briefly discuss the future extensions of this work. First, we will analyze whether the batch bound method of Algorithm 1 admits super-linear convergence rate and thus match the convergence rate of quasi-Newton techniques (the existing convergence analysis in the original paper shows linear rate only). On the empirical side, we will investigate applying the bound techniques discussed in this paper to optimize each layer of the network during backpropagation. Per-layer bounds, when combined together, may potentially lead to a universal quadratic bound on the original highly complex deep learning loss function. This approach would open up new possibilities for training deep learning models as it reduces the deep learning non-convex optimization problem to a convex one. We will also investigate applying the developed technique to backpropagation-free [16] setting and large-batch [46] training of deep learning models. Finally, we will explore the applicability of the bound techniques in the context of biasing the gradient to explore wide valleys in the non-convex optimization landscape [14]. In this case enforcing the width of the bound to be sufficiently large should provide a simple mechanism for finding solutions that lie in the flat regions of the landscape.

1. http://archive.ics.uci.edu/ml/datasets/Adult
2. https://pytorch.org/docs/stable/torchvision/datasets.html

References

[1] Galen Andrew and Jianfeng Gao. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pages 33–40, 2007.

[2] S. Balakrishnan, M. J. Wainwright, and B. Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *Ann. Statist.*, 45(1):77–120, 2017.

[3] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina. Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses. *Physical review letters*, 115(12):128101, 2015.

[4] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. In *PNAS*, 2016.

[5] C. Baldassi, F. Gerace, C. Lucibello, L. Saglietti, and R. Zecchina. Learning may need only a few bits of synaptic precision. *Physical Review E*, 93(5):052313, 2016.

[6] C. H. Bennett. Efficient Estimation of Free Energy Differences from Monte Carlo Data. *Journal of Computational Physics*, 22(2):245–268, October 1976. doi: 10.1016/0021-9991(76)90078-4.

[7] Steven J Benson and Jorge J More. A limited memory variable metric method in subspaces and bound constrained optimization problems. In *in Subspaces and Bound Constrained Optimization Problems*. Citeseer, 2001.

[8] Adam Berger. The improved iterative scaling algorithm: A gentle introduction, 1997.

[9] Dankmar Böhning and Bruce G Lindsay. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663, 1988.

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

[11] Charles George Broyden, John E Dennis Jr, and Jorge J Moré. On the local and superlinear convergence of quasi-newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

[12] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[13] Miguel Á. Carreira-Perpiñán and Geoffrey E. Hinton. On contrastive divergence learning. In *AISTATS*, 2005.

[14] P. Chaudhari, **A. Choromanska**, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017.

[15] P. Chaudhari, **A. Choromanska**, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys (journal version). *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[16] Anna Choromanska, Benjamin Cowen, Sadhana Kumaravel, Ronny Luss, Mattia Rigotti, Irina Rish, Paolo Diachille, Viatcheslav Gurev, Brian Kingsbury, Ravi Tejwani, and Djallel Bouneffouf. Beyond backprop: Online alternating minimization with auxiliary variables. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1193–1202. PMLR, 2019. URL http://proceedings.mlr.press/v97/choromanska19a.html.

[17] John N Darroch and Douglas Ratcliff. Generalized iterative scaling for log-linear models. *The annals of mathematical statistics*, pages 1470–1480, 1972.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1): 1–38, 1977.

[19] Y. Feng and Y. Tu. How neural networks find generalizable solutions: Self-tuned annealing in deep learning. *CoRR*, abs/2001.01678, 2020.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[21] Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13(null): 307–361, February 2012. ISSN 1532-4435.

[22] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. URL http://www.ncbi.nlm.nih.gov/pubmed/12180402.

[23] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[24] Fuchun Huang and Yosihiko Ogata. Generalized pseudo-likelihood estimates for markov random fields on lattice. *Annals of the Institute of Statistical Mathematics*, 54:1–18, 02 2002. doi: 10.1023/A:1016170102988.

[25] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, December 2005. ISSN 1532-4435.

[26] Aapo Hyvarinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5):2499–2512, 2007. URL https://EconPapers.repec.org/RePEc:eee:csdana:v:51:y:2007:i:5:p:2499-2512.

[27] C. Jarzynski. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, 78:2690–2693, Apr 1997. doi: 10.1103/PhysRevLett.78.2690. URL https://link.aps.org/doi/10.1103/PhysRevLett.78.2690.

[28] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Finding flatter minima with sgd. In *ICLR Workshop Track*, 2018.

[29] Tony Jebara and Anna Choromanska. Majorization for crfs and latent likelihoods. In *Advances in Neural Information Processing Systems*, pages 557–565, 2012.

[30] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. *CoRR*, abs/1912.02178, 2019.

[31] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017. URL http://arxiv.org/abs/1609.04836.

[32] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research).

[33] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[34] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.

[35] Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM J. on Optimization*, 25(2):829–855, April 2015. ISSN 1052-6234. doi: 10.1137/140957639. URL https://doi.org/10.1137/140957639.

[36] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[37] R.M. Neal. Annealed importance sampling. *Statistics and Computing*, 11, 01 2001.

[38] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[39] L. Sagun, U. Evci, V. Ugur Güney, Y. Dauphin, and L. Bottou. Empirical analysis of the hessian of over-parametrized neural networks. In *ICLR Workshop*, 2018.

[40] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 213–220, 2003.

[41] U. Simsekli, L. Sagun, and M. Gürbüzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. *CoRR*, abs/1901.06053, 2019.

[42] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1064–1071, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390290. URL https://doi.org/10.1145/1390156.1390290.

[43] Tijmen Tieleman and Geoffrey Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 1033–1040, 2009.

[44] Hanna Wallach. *Efficient training of conditional random fields*. PhD thesis, Citeseer, 2002.

[45] W. Wen, Y. Wang, F. Yan, C. Xu, Y. Chen, and H. Li. Smoothout: Smoothing out sharp minima for generalization in large-batch deep learning. *CoRR*, abs/1805.07898, 2018.

[46] Y. You, I. Gitman, and B. Ginsburg. Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888, 2017.

[47] Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1033–1040. MIT Press, 2002. URL http://papers.nips.cc/paper/2125-the-concave-convex-procedure-cccp.pdf.

[48] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

Supplementary material

Appendix A. Proof for Theorem 2: sub-linear convergence rate

The proof in this section inspires by [11, 12, 32]. We analysis the convergence rate of Stochastic version of Bound Majorization Method. It is easy to know that the objective function

$$L(\boldsymbol{\theta}) = \frac{1}{T}\sum_{t=1}^{T}\left[\log(Z_{\mathbf{x}_t}(\boldsymbol{\theta})) - \boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}_t}(\mathbf{y}_t)\right] + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$$

is strongly convex and and twice continuously differentiable, where $Z_{\mathbf{x}_t}(\boldsymbol{\theta}) = \sum_{i=1}^{n}\exp(\boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}_t}(i))$. Therfore, we can make the following assumption

Assumption 1. *Assume that*

(1) The objective function L is twice continuous differentiable.

(2) There exists $0 < \lambda_1 < \lambda_2$ such that for all $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$\lambda_1\mathbf{I} \prec \nabla^2 L(\boldsymbol{\theta}) \prec \lambda_2\mathbf{I}, \tag{A.1}$$

For stochastic partition function bound (SPFB) method, define η_t to be the learning rate, $\boldsymbol{\Sigma}_t$ to be the Hessian approximation and $f(\boldsymbol{\theta};\mathbf{x}_t) = \log(Z_t(\boldsymbol{\theta})) - \boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}_t}(\mathbf{y}_t) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$ to be the unbiased estimation of objective function $L(\boldsymbol{\theta})$ in each iteration, then the update for parameter $\boldsymbol{\theta}$ is

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t(\boldsymbol{\Sigma}_t + \lambda\mathbf{I})^{-1}\nabla f(\boldsymbol{\theta}^t;\mathbf{x}_t). \tag{A.2}$$

In order to analysis the convergence for SPFB, we add a condition that the variance of estimation $f(\boldsymbol{\theta})$ is bounded then construct the following assumption:

Assumption 2. *Assume that*

(1) The objective function L is twice continuous differentiable.

(2) There exists $0 < \lambda_1 < \lambda_2$ such that for all $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$\lambda_1\mathbf{I} \prec \nabla^2 L(\boldsymbol{\theta}) \prec \lambda_2\mathbf{I}, \tag{A.3}$$

(3) There exists a constant σ, such that for all $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$\mathbf{E}_{\mathbf{x}}[\|f(\boldsymbol{\theta};\mathbf{x})\|]^2 \leq \sigma^2 \tag{A.4}$$

Lemma 1. *For matrices $\{\boldsymbol{\Sigma}_t\}$ compute in **Algorithm** 1, there exists $0 < \mu_1 < \mu_2$ such that for all $\boldsymbol{\Sigma}_t$*

$$\mu_1\mathbf{I} \prec (\boldsymbol{\Sigma}_t + \lambda\mathbf{I})^{-1} \prec \mu_2\mathbf{I}. \tag{A.5}$$

Proof For easy notation, we omit the subscripts and demote $\mathbf{\Sigma}_t$ as $\mathbf{\Sigma}$. From **Algorithm** 1, the formulation for $\mathbf{\Sigma}$ is

$$\mathbf{\Sigma} = z_0\mathbf{I} + \sum_{i=1}^{n} \beta_i \mathbf{l}_i \mathbf{l}_i^T, \tag{A.6}$$

where $z_0 \to 0^+$ is the initialization of z in algorithm 1. Define $z_i = \sum_{k=1}^{i} \alpha_k$, we can compute the upper bound of β_i

$$\beta_i = \frac{\tanh\left(\frac{1}{2}\log(\alpha_i/z_i)\right)}{2\log(\alpha_i/z_i)} = \frac{1}{4} \cdot \frac{\tanh\left(\frac{1}{2}\log(\alpha_i/z_i)\right)}{\frac{1}{2}\log(\alpha_i/z_i)} \leq \frac{1}{4}, \tag{A.7}$$

we can also conclude that there exits k such that $\frac{1}{n} < \frac{\alpha_k}{z_k} < 1$, which imply

$$\beta_k \geq \frac{\tanh(\frac{1}{2}\log(\frac{1}{n}))}{2\log(\frac{1}{n})}.$$

Since $\mathbf{l}_{i+1} = (-\frac{\alpha_1}{z_i}\mathbf{x}, \cdots, \frac{\alpha_i}{z_i}\mathbf{x}, \mathbf{x}, 0, \cdots, 0)$, where $\mathbf{x}$ is the current observation, we have

$$(1+\frac{1}{n})\|\mathbf{x}\|^2 \leq \|\mathbf{l}_{i+1}\|^2 = \frac{\alpha_1^2 + \alpha_2^2 + \cdots \alpha_i^2}{(\alpha_1 + \alpha_2 + \cdots \alpha_i)^2}\|\mathbf{x}\|^2 + \|\mathbf{x}\|^2 \leq 2\|\mathbf{x}\|^2 \tag{A.8}$$

Therefore,

$$\|\mathbf{\Sigma}\|^2 \geq \beta_k \|\mathbf{l}_k\|^2 \geq \left(1+\frac{1}{n}\right)\frac{\tanh(\frac{1}{2}\log(\frac{1}{n}))}{2\log(\frac{1}{n})}\|\mathbf{x}\|^2. \tag{A.9}$$

Based on previous proof, the upperbound for matrix $\mathbf{\Sigma}$ is

$$\begin{aligned} \|\mathbf{\Sigma}\|^2 &\leq z_0^2 + \sum_{i=1}^{n} \beta_i \|\mathbf{l}_i \mathbf{l}_i^T\|^2 \leq z_0^2 + \frac{1}{4}\sum_{i=1}^{n} \|\mathbf{l}_i\|^4 \\ &\leq z_0^2 + \frac{1}{4}\|\mathbf{x}\|^4 \sum_{i=1}^{n} 2 \leq z_0^2 + \sqrt{\frac{n}{2}}\|\mathbf{x}\|^4. \end{aligned} \tag{A.10}$$

When $z_0 \to 0^+$, $\|\mathbf{\Sigma}\| \leq \sqrt{\frac{n}{2}}\|\mathbf{x}\|^2$.

Denote $\{\mathbf{x}_t\}$ as the set of all observations. Define $\mu_2 = 1/\left(\left(1+\frac{1}{n}\right)\frac{\tanh(\frac{1}{2}\log(\frac{1}{n}))}{2\log(\frac{1}{n})}\max_t \|\mathbf{x}_t\|^2 + \lambda\right)$, $\mu_1 = 1/\left(\sqrt{\frac{n}{2}}\max_t \|\mathbf{x}_t\|^2 + \lambda\right)$,

$$1/\mu_2 \leq \|\mathbf{\Sigma}_t + \lambda\mathbf{I}\| \leq 1/\mu_1 \Longrightarrow \mu_2 \geq \|(\mathbf{\Sigma}_t + \lambda\mathbf{I})^{-1}\| \geq \mu_1.$$

Theorem 2. *$\{\boldsymbol{\theta}^t\}$ is the sequence of parameters generated by Algorithm 2. There exists $0 < \mu_1 < \mu_2$, $0 < \lambda_1 < \lambda_2$ such that for all iterations t,*

$$\mu_1\mathbf{I} \prec (\mathbf{\Sigma}_t + \lambda\mathbf{I})^{-1} \prec \mu_2\mathbf{I} \quad \textit{and} \quad \lambda_1\mathbf{I} \prec \nabla^2 L(\boldsymbol{\theta}) \prec \lambda_2\mathbf{I}, \tag{A.11}$$

and there exists a constant σ, such that for all $\boldsymbol{\theta} \in \mathbb{R}^d$,

$$\mathbf{E}_{\mathbf{x}_t}[\|\|f(\boldsymbol{\theta}; \mathbf{x}_t)\|\|]^2 \leq \sigma^2. \tag{A.12}$$

Define the learning rate in iteration t as

$$\eta_t = \eta_0/t \qquad \textit{where } \eta_0 > 1/(2\mu_1\lambda_1).$$

Then for all $t > 1$,

$$\mathbf{E}[L(\boldsymbol{\theta}^t) - L(\boldsymbol{\theta}^*)] \leq Q(\eta_0)/t, \tag{A.13}$$

where $Q(\eta_0) = \max\left\{\frac{\lambda_2\mu_2^2\eta_0^2\sigma^2}{2(2\mu_1\lambda_1\eta_0-1)}, L(\boldsymbol{\theta}^1) - L(\boldsymbol{\theta}^*)\right\}$.

Proof

$$\begin{aligned} L(\boldsymbol{\theta}^{t+1}) &= L(\boldsymbol{\theta}^t - \eta_t(\boldsymbol{\Sigma}_t + \lambda\mathbf{I})^{-1}\nabla f(\boldsymbol{\theta}^t; \mathbf{x}_t)) \\ &\leq L(\boldsymbol{\theta}^t) + \nabla L(\boldsymbol{\theta}^t)^T\left(-\eta_t(\boldsymbol{\Sigma}_t + \lambda\mathbf{I})^{-1}\nabla f(\boldsymbol{\theta}^t; \mathbf{x}_t)\right) + \frac{\lambda_2}{2}\left\|\eta_t(\boldsymbol{\Sigma}_t + \lambda\mathbf{I})^{-1}\nabla f(\boldsymbol{\theta}^t; \mathbf{x}_t)\right\|^2 \\ &\leq L(\boldsymbol{\theta}^t) - \eta_t\nabla L(\boldsymbol{\theta}^t)^T(\boldsymbol{\Sigma}_t + \lambda\mathbf{I})^{-1}\nabla f(\boldsymbol{\theta}^t; \mathbf{x}_t) + \frac{\lambda_2}{2}\eta_t^2\mu_2^2\left\|\nabla f(\boldsymbol{\theta}^t; \mathbf{x}_t)\right\|^2. \end{aligned} \tag{A.14}$$

Taking the expectation, we have

$$\mathbf{E}[L(\boldsymbol{\theta}^{t+1})] \leq \mathbf{E}[L(\boldsymbol{\theta}^t)] - \eta_t\mu_1\left\|\nabla L(\boldsymbol{\theta}^t)\right\|^2 + \frac{\lambda_2}{2}\eta_t^2\mu_2^2\sigma^2. \tag{A.15}$$

Now, we want to correlate $\left\|\nabla L(\boldsymbol{\theta}^t)\right\|^2$ with $L(\boldsymbol{\theta}^\mathbf{t}) - L(\boldsymbol{\theta}^*)$. For all $\mathbf{v} \in \mathbb{R}^d$, by condition (2) in Assumption 2,

$$\begin{aligned} L(\mathbf{v}) &\geq L(\boldsymbol{\theta}^t) + \nabla L(\boldsymbol{\theta}^t)^T(\mathbf{v} - \boldsymbol{\theta}^t) + \frac{\lambda_1}{2}\|\mathbf{v} - \boldsymbol{\theta}^t\|^2 \\ &\geq L(\boldsymbol{\theta}^t) - \nabla L(\boldsymbol{\theta}^t)^T\left(\frac{1}{\lambda_1}\nabla L(\boldsymbol{\theta}^t)\right) + \frac{\lambda_1}{2}\|\frac{1}{\lambda_1}\nabla L(\boldsymbol{\theta}^t)\|^2 \\ &= L(\boldsymbol{\theta}^t) - \frac{1}{2\lambda_1}\|\nabla L(\boldsymbol{\theta}^t)\|^2, \end{aligned} \tag{A.16}$$

the second inequality comes from computing the minimum of quadratic function $q(\mathbf{v}) = L(\boldsymbol{\theta}^k) + \nabla L(\boldsymbol{\theta}^k)^T(\mathbf{v} - \boldsymbol{\theta}^k) + \frac{\lambda_1}{2}\|\mathbf{v} - \boldsymbol{\theta}^k\|^2$. Setting $\mathbf{v} = \boldsymbol{\theta}^*$ yields

$$2\lambda_1[L(\boldsymbol{\theta}^\mathbf{t}) - L(\boldsymbol{\theta}^*)] \leq \|\nabla L(\boldsymbol{\theta}^t)\|^2, \tag{A.17}$$

which together with formula (A.15) yields

$$\begin{aligned} \mathbf{E}[L(\boldsymbol{\theta}^{t+1}) - L(\boldsymbol{\theta}^*)] &\leq \mathbf{E}[L(\boldsymbol{\theta}^t) - L(\boldsymbol{\theta}^*)] - 2\eta_t\mu_1\lambda_1\mathbf{E}[L(\boldsymbol{\theta}^t) - L(\boldsymbol{\theta}^*)] + \frac{\lambda_2}{2}\eta_t^2\mu_2^2\sigma^2 \\ &= (1 - 2\eta_t\mu_1\lambda_1)\mathbf{E}[L(\boldsymbol{\theta}^t) - L(\boldsymbol{\theta}^*)] + \frac{\lambda_2}{2}\eta_t^2\mu_2^2\sigma^2. \end{aligned} \tag{A.18}$$

Define $\phi_t = \mathbf{E}[L(\boldsymbol{\theta}^t) - L(\boldsymbol{\theta}^*)]$, $Q(\eta_0) = \max\left\{\frac{\lambda_2\mu_2^2\eta_0^2\sigma^2}{2(2\mu_1\lambda_1\eta_0-1)}, L(\boldsymbol{\theta}^1) - L(\boldsymbol{\theta}^*)\right\}$, when $t = 1$, $\phi_1 = L(\boldsymbol{\theta}^1) - L(\boldsymbol{\theta}^*) = Q(\eta_0)/1$ holds. We finish the proof by induction. Assume $\phi_t \leq \frac{Q(\eta_0)}{t}$,

$$\begin{aligned}
\phi_{t+1} &\leq (1 - 2\eta_t\mu_1\lambda_1)\phi_t + \frac{\lambda_2}{2}\eta_t^2\mu_2^2\sigma^2 \\
&= (1 - \frac{2\eta_0\mu_1\lambda_1}{t})\frac{Q(\eta_0)}{t} + \frac{\lambda_2\eta_0^2\mu_2^2\sigma^2}{2t^2} \\
&= \frac{t - 2\eta_0\mu_1\lambda_1}{t^2}Q(\eta_0) + \frac{\lambda_2\eta_0^2\mu_2^2\sigma^2}{2t^2} \\
&= \frac{t-1}{t^2}Q(\eta_0) - \frac{2\eta_0\mu_1\lambda_1 - 1}{t^2}Q(\eta_0) + \frac{\lambda_2\eta_0^2\mu_2^2\sigma^2}{2t^2} \\
&\leq \frac{t-1}{t^2}Q(\eta_0) - \frac{2\eta_0\mu_1\lambda_1 - 1}{t^2}\frac{\lambda_2\mu_2^2\eta_0^2\sigma^2}{2(2\mu_1\lambda_1\eta_0 - 1)} + \frac{\lambda_2\eta_0^2\mu_2^2\sigma^2}{2t^2} \\
&\leq \frac{Q(\eta_0)}{t+1}
\end{aligned} \tag{A.19}$$

Appendix B. Proof for Thm 3: Low-rank Bound

The lower-bound in [29] is not correct since $\mathbf{a} \perp \mathbf{g}$ is not sufficient for $\mathbf{x}^T(\mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T)\mathbf{x} = 0, \forall \mathbf{x}$. We loose and fix the bound in this section. We use some proof steps in [29].

Lemma 2. *$\forall \mathbf{x} \in \mathbb{R}^d$, $\forall \mathbf{l} \in \mathbb{R}^{+d}(l \geq 0)$, we have*

$$\sum_{i=1}^{d} \mathbf{x}^2(i)\mathbf{l}(i) \geq \left(\sum_{i=1}^{d} \mathbf{x}(i)\frac{\mathbf{l}(i)}{\sqrt{\sum_{j=1}^{d}\mathbf{l}(j)}}\right)$$

Proof By Jensen's inequality, if $f : \mathbb{R} \to \mathbb{R}$ convex, $\{a_i\}_{i=1}^d$ satisfies $\sum_{i=1}^d a_i = 1$, then

$$\sum_{i=1}^{d} a_i f(\mathbf{x}(i)) \geq f\left(\sum_{i=1}^{d} a_i \mathbf{x}(i)\right)$$

Set $a_i = \frac{\mathbf{l}(i)}{\sum_{i=1}^d \mathbf{l}(i)}$, $f(x) = x^2$,

$$\sum_{i=1}^{d} \mathbf{x}(i)^2\frac{\mathbf{l}(i)}{\sum_{i=1}^{d}\mathbf{l}(i)} \geq \left(\sum_{i=1}^{d} \mathbf{x}(i)\frac{\mathbf{l}(i)}{\sum_{i=1}^{d}\mathbf{l}(i)}\right)^2$$

$$\sum_{i=1}^{d} \mathbf{x}^2(i)\mathbf{l}(i) \geq \left(\sum_{i=1}^{d} \mathbf{x}(i)\frac{\mathbf{l}(i)}{\sqrt{\sum_{j=1}^{d}\mathbf{l}(j)}}\right)$$

Lemma 3. *If $\mathbf{a}, \mathbf{g} \in \mathbb{R}^n$ non-zero, then $rank(\mathbf{a}\mathbf{g}^T) = 1$.*

Proof

$$rank(\mathbf{a}\mathbf{g}^T) = \min\{rank(\mathbf{a}), rank(\mathbf{g})\} = 1$$

Lemma 4. *Let* $\mathbf{a}, \mathbf{g} \in \mathbb{R}^n$ *non-zero, matrix* $\mathbf{A} = \mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T \in \mathbb{R}^{n\times n}$. *If* $\mathbf{a} \perp \mathbf{g}$ *and* $\mathbf{A} \neq \mathbf{0}$, *then* $\boldsymbol{A}$ *has exactly 2 opposite eigenvalues* $\pm\|\mathbf{a}\|_2\|\mathbf{g}\|_2$.

Proof Because $rank(\mathbf{a}\mathbf{g}^T) = rank(\mathbf{g}\mathbf{a}^T) = 1$ (By **Lemma** 3),

$$rank(\mathbf{A}) = rank(\mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T) \leq rank(\mathbf{a}\mathbf{g}^T) + rank(\mathbf{g}\mathbf{a}^T) = 2.$$

(a) $rank(\mathbf{A}) = 1$, there is only one non-zero eigenvalue λ.

$$\lambda = trace(\mathbf{A}) = \sum_{i=1}^{n} \mathbf{A}_{ii} = \sum_{i=1}^{n} \mathbf{a}_i\mathbf{g}_i + \mathbf{a}_i\mathbf{g}_i = 2\mathbf{a}^T\mathbf{g} = 0.$$

Therefore, all eigenvalues of matrix $\mathbf{A}$ are 0 and $\mathbf{A} = \mathbf{0}$, which contradicts the condition $\mathbf{A} \neq 0$.

(b) $rank(\mathbf{A}) = 2$, assume λ_1, λ_2 are 2 non-zero eigenvalues of $\mathbf{A}$.

$$\lambda_1 + \lambda_2 = trace(\mathbf{A}) = 0 \tag{B.1}$$

Without loss of generality, assume $\lambda_1 > 0$, then the characteristic polynomial of matrix A is

$$\lambda^{n-2}(\lambda - \lambda_1)(\lambda - \lambda_2) = \lambda^{n-2}(\lambda^2 - \lambda_1^2),$$

and matrix A satisfies

$$\mathbf{A}^{n-2}(\mathbf{A}^2 - \lambda_1^2\mathbf{I}) = \mathbf{0} \tag{B.2}$$

Because

$$\begin{aligned}
\mathbf{A}^3 = \mathbf{A}^2\mathbf{A} &= (\mathbf{a}\mathbf{g}^T\mathbf{a}\mathbf{g}^T + \mathbf{a}\mathbf{g}^T\mathbf{g}\mathbf{a}^T + \mathbf{g}\mathbf{a}^T\mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T\mathbf{g}\mathbf{a}^T)(\mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T) \\
&= (\|\mathbf{a}\|_2^2\mathbf{g}\mathbf{g}^T + \|\mathbf{g}\|_2^2\mathbf{a}\mathbf{a}^T)(\mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T) \\
&= \|\mathbf{a}\|_2^2\|\mathbf{g}\|_2^2(\mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T) \\
&= \|\mathbf{a}\|_2^2\|\mathbf{g}\|_2^2\mathbf{A},
\end{aligned}$$

We can conclude that

$$\mathbf{A}(\mathbf{A}^2 - \|\mathbf{a}\|_2^2\|\mathbf{g}\|_2^2\mathbf{I}) = 0 \Longrightarrow \mathbf{A}^{n-2}(\mathbf{A}^2 - \|\mathbf{a}\|_2^2\|\mathbf{g}\|_2^2\mathbf{I}) = 0. \tag{B.3}$$

From formula (B.1), (B.2) and (B.3) we can conclude that the eigenvalues $\lambda_1 = \|\mathbf{a}\|_2\|\mathbf{g}\|_2$ and $\lambda_2 = -\|\mathbf{a}\|_2\|\mathbf{g}\|_2$.

Theorem 3. *Let* $Z_{\mathbf{x}}(\boldsymbol{\theta}) = \sum_{y=1}^{n} \exp\left(\boldsymbol{\theta}^T\mathbf{f}_{\mathbf{x}}(y)\right)$. *In each iteration of the* x*-loop in Algorithm 3 finds* $z, \boldsymbol{\mu}, \mathbf{V}, \mathbf{S}, \mathbf{D}$ *such that*

$$Z_{\mathbf{x}}(\boldsymbol{\theta}) \leq z \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T(\mathbf{V}^T\mathbf{S}\mathbf{V} + \mathbf{D})(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T\boldsymbol{\mu}\right) \tag{B.4}$$

Proof Define $Z^i_{\mathbf{x}}(\boldsymbol{\theta}) = \sum_{y=1}^{i} \exp(\boldsymbol{\theta}^T \mathbf{f}_{\mathbf{x}}(y))$, then the partition function is represented as $Z_{\mathbf{x}}(\boldsymbol{\theta}) = Z^n_{\mathbf{x}}(\boldsymbol{\theta})$, where n is the number of labels. From proof of Thm 1 (details included in [29]), we already find a sequence of matrix $\{\boldsymbol{\Sigma}_i\}_{i=1}^n$, vector $\{\boldsymbol{\mu}\}_{i=1}^n$ and constant $\{z_i\}_{i=1}^n$ such that:

$$Z^i_{\mathbf{x}}(\boldsymbol{\theta}) \leq z_i \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}_i (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \boldsymbol{\mu}_i\right), \tag{B.5}$$

where the terminate terms $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_n$, $\boldsymbol{\mu} = \boldsymbol{\mu}_n$, $z = z_n$ are the output of algorithm 1. If we could find sequences of $\{\mathbf{V}_i\}_{i=1}^n \subset \mathbb{R}^{k\times d}$ (orthnormal), $\{\mathbf{S}_i\}_{i=1}^n \subset \mathbb{R}^{k\times k}$ and $\{\mathbf{D}_i\}_{i=1}^n \subset \mathbb{R}^{d\times d}$ (diagonal) upper-bounds matrices $\{\boldsymbol{\Sigma}_i\}_{i=1}^n \subset \mathbb{R}^{d\times d}$ as

$$\mathbf{x}^T \boldsymbol{\Sigma}_i \mathbf{x} \leq \mathbf{x}^T (\mathbf{V}_i^T \mathbf{S}_i \mathbf{V}_i + \mathbf{D}_i)\mathbf{x} \quad \forall \mathbf{x} \in \mathbb{R}^d, \tag{B.6}$$

the upper-bound B.5 of partition function over only i labels can be renewed as

$$Z^i_{\mathbf{x}}(\boldsymbol{\theta}) \leq z_i \exp\left(\frac{1}{2}(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \left(\mathbf{V}_i^T \mathbf{S}_i \mathbf{V}_i + \mathbf{D}_i\right)(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \boldsymbol{\mu}_i\right). \tag{B.7}$$

When $i = n$, denote $\mathbf{V}, \mathbf{S}, \mathbf{D}, \boldsymbol{\mu}, z = \mathbf{V}_n, \mathbf{S}_n, \mathbf{D}_n, \boldsymbol{\mu}_n, z_n$, the formula $B.7$ is equivalent to $B.23$ and we finish the proof. We successfully decompose $\boldsymbol{\Sigma}_i$ into lower-rank while keep the upper-bound at the same time. In the following part, we are going to show how to construct the matrix sequences $\{\mathbf{V}_i\}_{i=1}^n$, $\{\mathbf{S}_i\}_{i=1}^n$ and $\{\mathbf{D}_i\}_{i=1}^n$ satisfy the condition B.6. The proof is based on mathematical induction.

- From proof of Thm 1 in [29], $z_0 \to 0^+$, $\boldsymbol{\Sigma}_0 = z_0\mathbf{I}$. Define $\mathbf{V}_0 = \mathbf{0}$, $\mathbf{S}_0 = \mathbf{0}$ and $\mathbf{D}_0 = z_0\mathbf{I}$, it is obvious that $\mathbf{x}^T\boldsymbol{\Sigma}_0\mathbf{x} = \mathbf{x}^T(\mathbf{V}_0^T\mathbf{S}_0\mathbf{V}_0 + \mathbf{D}_0)\mathbf{x}$ holds for all $\mathbf{x} \in \mathbb{R}^d$.
- Assume $\mathbf{V}_{i-1}$, $\mathbf{S}_{i-1}$ and $\mathbf{D}_{i-1}$ satisfy condition (B.6), we are going to find $\mathbf{V}_i$, $\mathbf{S}_i$ and $\mathbf{D}_i$ satisfy this condition as well. From line 5 in algorithm 1, $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i-1} + \mathbf{r}\mathbf{r}^T$, where $\mathbf{r} = \sqrt{\beta}\mathbf{l}$. Define the subspace constructed by row vectors of $\mathbf{V}_{i-1}$ as $\mathcal{A} = span\{\mathbf{V}_{i-1}(1,\cdot), ..., \mathbf{V}_{i-1}(k,\cdot)\}$. Map the vector $\mathbf{r}$ onto subspace $\mathcal{A}$ and denote the residual orthogonal to subspace $\mathcal{A}$ as $\mathbf{g}$

$$\begin{aligned}\mathbf{r} &= \sum_{j=1}^{d} \mathbf{r}\mathbf{V}_{i-1}(j,\cdot)\mathbf{V}_{i-1}^T(j,\cdot) + \mathbf{g} \\ &= \mathbf{V}_{i-1}^T\mathbf{V}_{i-1}\mathbf{r} + \mathbf{g},\end{aligned} \tag{B.8}$$

substitute (B.8) into $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i-1} + \mathbf{r}\mathbf{r}^T$, we have

$$\begin{aligned}\boldsymbol{\Sigma}_i &= \boldsymbol{\Sigma}_{i-1} + (\mathbf{V}_{i-1}^T\mathbf{V}_{i-1}\mathbf{r} + \mathbf{g})(\mathbf{V}_{i-1}^T\mathbf{V}_{i-1}\mathbf{r} + \mathbf{g})^T \\ &= \mathbf{V}_{i-1}^T(\mathbf{S}_{i-1} + \mathbf{V}_{i-1}\mathbf{r}\mathbf{r}_i^T\mathbf{V}_{i-1}^T)\mathbf{V}_{i-1} + \mathbf{D}_{i-1} + \mathbf{g}\mathbf{g}^T + \mathbf{V}_{i-1}^T\mathbf{V}_{i-1}\mathbf{r}\mathbf{g}^T + \mathbf{g}\mathbf{r}_i^T\mathbf{V}_{i-1}^T\mathbf{V}_{i-1}.\end{aligned} \tag{B.9}$$

Define $\mathbf{a} = \mathbf{V}_{i-1}^T\mathbf{V}_{i-1}\mathbf{r}$, Equation (B.9) can be simplified as

$$\boldsymbol{\Sigma}_i = \mathbf{V}_{i-1}^T(\mathbf{S}_{i-1} + \mathbf{V}_{i-1}\mathbf{r}\mathbf{r}^T\mathbf{V}_{i-1}^T)\mathbf{V}_{i-1} + \mathbf{D}_{i-1} + \mathbf{g}\mathbf{g}^T + \mathbf{a}\mathbf{g}^T + \mathbf{g}\mathbf{a}^T, \tag{B.10}$$

the corresponding quadratic form is

$$\mathbf{x}^T\mathbf{\Sigma}_i\mathbf{x} = \mathbf{x}^T\left[\mathbf{V}_{i-1}^T(\mathbf{S}_{i-1}+\mathbf{V}_{i-1}\mathbf{r}\mathbf{r}^T\mathbf{V}_{i-1}^T)\mathbf{V}_{i-1}+\mathbf{D}_{i-1}+\mathbf{g}\mathbf{g}^T\right]+\mathbf{x}^T(\mathbf{a}\mathbf{g}^T+\mathbf{g}\mathbf{a}^T)\mathbf{x}. \tag{B.11}$$

By **Lemma** 4, the maximum eigenvalue of matrix $\mathbf{a}\mathbf{g}^T+\mathbf{g}\mathbf{a}^T$ is $\|\mathbf{a}\|_2\|\mathbf{g}\|_2$. Define $\mathbf{P} = \|\mathbf{a}\|_2\|\mathbf{g}\|_2 I$, we have

$$\mathbf{x}^T(\mathbf{a}\mathbf{g}^T+\mathbf{g}\mathbf{a}^T)\mathbf{x} \le \max\{\lambda|\lambda \text{ is eigenvalue of } A\}\mathbf{x}^T\mathbf{x} = \|\mathbf{a}\|_2\|\mathbf{g}\|_2\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{P}\mathbf{x}. \tag{B.12}$$

Define

$$\begin{aligned}\mathbf{D}'_{i-1} &= \mathbf{D}_{i-1}+\mathbf{P}\in\mathbb{R}^{d\times d} \text{ (diagonal)},\\ \widetilde{\mathbf{\Sigma}_i} &:= \mathbf{V}_{i-1}^T(\mathbf{S}_{i-1}+\mathbf{V}_{i-1}\mathbf{r}_i\mathbf{r}_i^T\mathbf{V}_{i-1}^T)\mathbf{V}_{i-1}+\mathbf{D}'_{i-1}+\mathbf{g}\mathbf{g}^T,\end{aligned} \tag{B.13}$$

by (B.11) and (B.12), it is easy to know,

$$\mathbf{x}^T\mathbf{\Sigma}_i\mathbf{x} \le \mathbf{x}^T\widetilde{\mathbf{\Sigma}_i}x \tag{B.14}$$

We have already prove $x^T\widetilde{\mathbf{\Sigma}_i}x$ is larger than $x^T\mathbf{\Sigma}_i x$, then all we need is the upper-bound of $x^T\widetilde{\mathbf{\Sigma}_i}x$. Perform SVD decomposition on $\mathbf{S}_{i-1}+\mathbf{V}_{i-1}\mathbf{r}\mathbf{r}^T\mathbf{V}_{i-1}^T$ and denote the result as

$$\mathbf{Q}_{i-1}^T\mathbf{S}'_{i-1}\mathbf{Q}_{i-1} = svd(\mathbf{S}_{i-1}+\mathbf{V}_{i-1}\mathbf{r}\mathbf{r}^T\mathbf{V}_{i-1}^T),$$

define $\mathbf{V}'_{i-1} = \mathbf{Q}_{i-1}\mathbf{V}_{i-1}$, then (B.13) can be simplified as

$$\begin{aligned}\widetilde{\mathbf{\Sigma}_i} &= \mathbf{V}_{i-1}^T\mathbf{Q}_{i-1}^T\mathbf{S}'_{i-1}\mathbf{Q}_{i-1}\mathbf{V}_{i-1}+\mathbf{D}'_{i-1}+\mathbf{g}\mathbf{g}^T\\ &= {\mathbf{V}'_{i-1}}^T\mathbf{S}'_{i-1}\mathbf{V}'_{i-1}+\mathbf{D}'_{i-1}+\mathbf{g}\mathbf{g}^T\\ &= \underbrace{\begin{bmatrix}{\mathbf{V}'_{i-1}}^T & \frac{\mathbf{g}}{\|\mathbf{g}\|_2}\end{bmatrix}\begin{bmatrix}\mathbf{S}'_{i-1} & \mathbf{0}^T\\ \mathbf{0} & \|\mathbf{g}\|_2^2\end{bmatrix}\begin{bmatrix}\mathbf{V}'_{i-1}\\ \frac{\mathbf{g}^T}{\|\mathbf{g}\|_2}\end{bmatrix}}_{=:\mathbf{B}}+\mathbf{D}'_{i-1}.\end{aligned} \tag{B.15}$$

It is easy to know that $rank(\mathbf{B}) = k+1$ and $\mathbf{B}$ is a $(k+1)$-svd decomposition. In order to keep the construction of $\mathbf{\Sigma}_i$, we have no choice but to remove the smallest eigenvalue and corresponding eigenvector from matrix $\mathbf{B}$.

case 1) $\|\mathbf{g}\|_2^2 \le \arg\min_{j=1\ldots k}\mathbf{S}'_{i-1}(j,j)$, remove eigenvalue $\|\mathbf{g}\|_2^2$ and corresponding eigenvector $\frac{\mathbf{g}}{\|\mathbf{g}\|_2}$.

$$\mathbf{\Sigma}'_i = {\mathbf{V}'_{i-1}}^T\mathbf{S}'_{i-1}\mathbf{V}'_{i-1}\mathbf{D}'_{i-1} = \widetilde{\mathbf{\Sigma}_i}-\mathbf{g}\mathbf{g}^t = \widetilde{\mathbf{\Sigma}_i}-c\mathbf{v}\mathbf{v}^T, \tag{B.16}$$

where $c = \|\mathbf{g}\|_2^2$,$\mathbf{v} = \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$. In this case $\mathbf{V}_i = \mathbf{V}'_{i-1}$, $\mathbf{S}_i = \mathbf{S}'_{i-1}$.

case 2) $\|\mathbf{g}\|_2^2 > \arg\min_{j=1\ldots k} \mathbf{S}'_{i-1}(j,j)$, remove m^{th} (absolute value smallest) eigenvalue in $\mathbf{S}'_{i-1}$ and corresponding eigenvalue $\mathbf{V}'_{i-1}(m,\cdot)$.

$$\begin{aligned}\boldsymbol{\Sigma}'_i &= {\mathbf{V}'_{i-1}}^T \mathbf{S}'_{i-1} \mathbf{V}'_{i-1} + \mathbf{D}'_{i-1} + \mathbf{g}\mathbf{g}^t - \mathbf{S}'_{i-1}(m,m)\mathbf{V}_{i-1}(m,\cdot){\mathbf{V}_{i-1}(m,\cdot)}^T \\ &= \widetilde{\boldsymbol{\Sigma}_i} - c\mathbf{v}\mathbf{v}^T, \end{aligned} \tag{B.17}$$

where $c = \mathbf{S}'_{i-1}(m,m)$,$\mathbf{v} = \mathbf{V}_{i-1}(m,\cdot)$. In this case,

$$\mathbf{V}_i = \begin{bmatrix} \mathbf{V}'_{i-1}(1,\cdot) \\ \ldots \\ \mathbf{V}'_{i-1}(m-1,\cdot) \\ \frac{\mathbf{g}}{\|\mathbf{g}\|_2^2} \\ \ldots \\ \mathbf{V}'_{i-1}(k,\cdot) \end{bmatrix},$$

$$\mathbf{S}_i = diag(\mathbf{S}'_{i-1}(1,1), \cdots, \mathbf{S}'_{i-1}(m-1,m-1), \|\mathbf{g}\|_2^2, \cdots, \mathbf{S}'_{i-1}(k,k)).$$

In both cases

$$\boldsymbol{\Sigma}'_i = \mathbf{V}_i^T \mathbf{S}_i \mathbf{V}_i + \mathbf{D}'_{i-1} \tag{B.18}$$

$$\widetilde{\boldsymbol{\Sigma}_i} = \boldsymbol{\Sigma}'_i + c\mathbf{v}\mathbf{v}^T, \tag{B.19}$$

where $\mathbf{V}_i \in \mathbb{R}^{k\times d}$ orthnormal, $\mathbf{S}_i \in \mathbb{R}^{k\times k}$ and $\mathbf{D}'_{i-1} \in \mathbb{R}^{d\times d}$ are diagonal, $\mathbf{v} \in \mathbb{R}^d$ and $\mathbf{v}^T\mathbf{v} = 1$. Therefore, for all $\mathbf{x} \in \mathbb{R}^d$ we have

$$\mathbf{x}^T\widetilde{\boldsymbol{\Sigma}_i}\mathbf{x} = \mathbf{x}^T\boldsymbol{\Sigma}'_i\mathbf{x} + c\mathbf{x}^T\mathbf{v}\mathbf{v}^T\mathbf{x} = \mathbf{x}^T\boldsymbol{\Sigma}'_i\mathbf{x} + c(\mathbf{x}^T\mathbf{v})^2, \tag{B.20}$$

$\boldsymbol{\Sigma}'_i$ is not sufficient for the condition $\mathbf{x}^T\boldsymbol{\Sigma}'_i\mathbf{x} \geq \mathbf{x}^T\widetilde{\boldsymbol{\Sigma}_i}\mathbf{x}(\forall\mathbf{x})$, we would like to relax it by constructing $\boldsymbol{\Sigma}''_i = \boldsymbol{\Sigma}'_i + \mathbf{F}$ ($\mathbf{F} \in \mathbb{R}^{d\times d}$ is diagonal) such that $\mathbf{x}^T\boldsymbol{\Sigma}''_i\mathbf{x} \geq \mathbf{x}^T\widetilde{\boldsymbol{\Sigma}_i}\mathbf{x}(\forall\mathbf{x})$, which is equivalent to

$$\begin{aligned} \mathbf{x}^T\boldsymbol{\Sigma}''_i\mathbf{x} &\geq \mathbf{x}^T\widetilde{\boldsymbol{\Sigma}_i}\mathbf{x} \\ \mathbf{x}^T(\boldsymbol{\Sigma}'_i + \mathbf{F})\mathbf{x} &\geq \mathbf{x}^T(\boldsymbol{\Sigma}'_i + c\mathbf{v}\mathbf{v}^T)\mathbf{x} \\ \sum_{i=1}^d \mathbf{x}^2(i)\mathbf{F}(i) &\geq c\left(\sum_{i=1}^d \mathbf{x}(i)\mathbf{v}(i)\right)^2. \end{aligned} \tag{B.21}$$

Set $\mathbf{F}(i) = c|\mathbf{v}(i)| \sum_{j=1}^{d} |\mathbf{v}(j)|$, by Lemma 2 we have

$$\begin{aligned}\sum_{i=1}^{d} \mathbf{x}^2(i)\mathbf{F}(i) &= \sum_{i=1}^{d} |\mathbf{x}(i)|^2 \mathbf{F}(i) \\ &\geq \left(\sum_{i=1}^{d} |\mathbf{x}(i)| \frac{\mathbf{F}(i)}{\sqrt{\sum_{j=1}^{d} \mathbf{F}(i)}} \right) \\ &= \left(\sum_{i=1}^{d} |\mathbf{x}(i)| \frac{c|\mathbf{v}(i)| \sum_{j=1}^{d} |\mathbf{v}(j)|}{\sqrt{c} \sum_{j=1}^{d} c|\mathbf{v}(i)|} \right)^2 \\ &\geq c \left(\sum_{i=1}^{d} \mathbf{x}(i)\mathbf{v}(i) \right)^2 \end{aligned} \tag{B.22}$$

Combine (B.21) and (B.22), $\mathbf{\Sigma}_i'' = \mathbf{\Sigma}_i' + \mathbf{F}$ is what we want. Let $\mathbf{D}_i = \mathbf{D}_{i-1}' + \mathbf{F}$, together with $\mathbf{V}_i$ and $\mathbf{S}_i$ defined in former case 1) and case 2), we have

$$\mathbf{x}^T \mathbf{\Sigma}_i \mathbf{x} \leq \mathbf{x}^T \widetilde{\mathbf{\Sigma}}_i \mathbf{x} \leq \mathbf{x}^T \mathbf{\Sigma}_i'' \mathbf{x} = \mathbf{x}^T \left(\mathbf{V}_i^T \mathbf{S}_i \mathbf{V}_i + \mathbf{D}_i \right) \mathbf{x}$$

We already construct sequences of matrix $\{\mathbf{V}_i\}_{i=1}^n$, $\{\mathbf{S}_i\}_{i=1}^n$ and $\{\mathbf{D}_i\}_{i=1}^n$ satisfies the condition (B.6). Define $\mathbf{V}, \mathbf{S}, \mathbf{D} = \mathbf{V}_n, \mathbf{S}_n, \mathbf{D}_n$, keep $\boldsymbol{\mu}$ and z the same as what they are in Thm 1, the formula (B.7) implies

$$Z_{\mathbf{x}}(\boldsymbol{\theta}) \leq z \exp\left(\frac{1}{2} (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T (\mathbf{V}^T \mathbf{S} \mathbf{V} + \mathbf{D}) (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}})^T \boldsymbol{\mu} \right) \tag{B.23}$$