Truncated Inference for Latent Variable Optimization Problems: Application to Robust Estimation and Learning

Christopher Zach[0000−0003−2840−6187] and Huu Le[0000−0001−7562−7180]

Chalmers University of Technology, Gothenburg, Sweden
{zach,huul}@chalmers.se

Abstract. Optimization problems with an auxiliary latent variable structure in addition to the main model parameters occur frequently in computer vision and machine learning. The additional latent variables make the underlying optimization task expensive, either in terms of memory (by maintaining the latent variables), or in terms of runtime (repeated exact inference of latent variables). We aim to remove the need to maintain the latent variables and propose two formally justified methods, that dynamically adapt the required accuracy of latent variable inference. These methods have applications in large scale robust estimation and in learning energy-based models from labeled data.

Keywords: Majorization-minimization, latent variable models, stochastic gradient methods

1 Introduction

In this work[1] we are interested in optimization problems that involve additional latent variables and therefore have the general form,

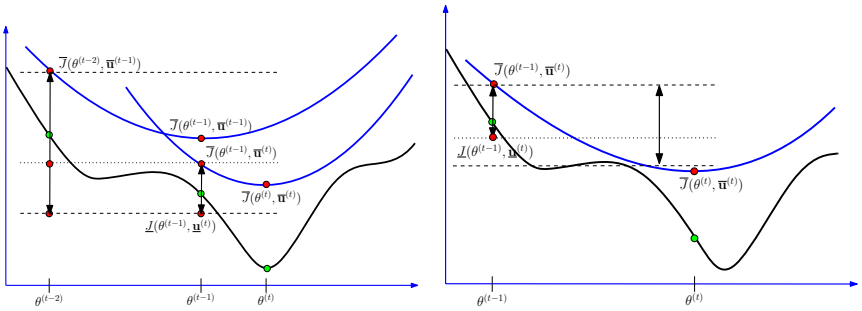
$$\min_{\theta} \min_{\overline{\mathbf{u}}} \overline{J}(\theta, \overline{\mathbf{u}}) =: \min_{\theta} J(\theta), \tag{1}$$

where θ are the main parameters of interest and $\overline{\mathbf{u}}$ denote the complete set of latent variables. By construction $\overline{J}(\theta, \overline{\mathbf{u}})$ is always an upper bound to the "ideal" objective J. In typical computer vision and machine learning settings the objective function in Eq. 1 has a more explicit structure as follows,

$$\overline{J}(\theta, \overline{\mathbf{u}}) = \frac{1}{N} \sum\nolimits_{i=1}^{N} \overline{J}_i(\theta, \overline{u}_i), \tag{2}$$

where the index i ranges over e.g. training samples or over observed measurements. Each $\overline{u}_i$ corresponds to the inferred (optimized) latent variable for each term, and $\overline{\mathbf{u}}$ is the entire collection of latent variables, i.e. $\overline{\mathbf{u}} = (\overline{u}_1, \ldots, \overline{u}_N)$. Examples for this problem class are models for (structured) prediction with latent

[1] This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.



(a) Relaxed generalized MM (b) Sufficient descent MM

Fig. 1. Illustration of the principle behind our proposed majorization-minimization variants. Left: relaxed generalized MM requires that the current duality gap at $\theta^{(t-1)}$ (between dotted and lower dashed lines) is at most a given fraction of the gap induced by the previous upper bound (between dashed lines). Right: sufficient descent MM requires that the current duality gap (between upper dashed and dotted lines) is at most a given fraction of a guaranteed decrease (between dashed lines).

variables [10, 32], supervised learning of energy-based models [20, 30] (in both scenarios N labeled training samples are provided), and robust estimation using explicit confidence weights [11, 35] (where N corresponds to the number of sensor measurements).

We focus on the setting when N is very large, and maintaining the values of $\overline{u}_i$ for all N terms in memory is intractable. In particular, storing the entire vector $\overline{\mathbf{u}}$ is undesirable when the dimensionality of each $\overline{u}_i$ is large. In one of our applications $\overline{u}_i$ represents the entire set of unit activations in a deep neural network, and therefore $\overline{u}_i$ is high-dimensional in such cases.

Observe that neither $J(\theta)$ nor $\nabla J(\theta)$ are easy to evaluate directly. By using a variable projection approach, the loss J in Eq. 2 can in principle be optimized using a "state-less" gradient method,

$$\nabla J(\theta) = \frac{1}{N} \sum\nolimits_{i=1}^{N} \nabla_\theta \overline{J}_i(\theta; \overline{u}_i^*(\theta)) \tag{3}$$

where $\overline{u}_i^*(\theta) = \arg\min_{\overline{u}_i} \overline{J}_i(\theta; \overline{u}_i)$. Usually determining $\overline{u}_i^*(\theta)$ requires itself an iterative minimization method, hence exactly solving $\arg\min_{\overline{u}_i} \overline{J}_i(\theta; \overline{u}_i)$ renders the computation of $\nabla J(\theta)$ expensive in terms of run-time (e.g. it requires solving a quadratic program in the application presented in Section 6.2). On the other hand, by using Eq. 3 there is no need to explicitly keep track of the values $\overline{u}_i^*(\theta)$ (as long as determining the minimizer $\overline{u}_i^*(\theta)$ is "cold-started", i.e. run from scratch). Note that Eq. 3 is only correct for stationary points $\overline{u}_i^*(\theta)$. For inexact minimizers $\overline{u}_i'(\theta) \approx \overline{u}_i^*(\theta)$ the second term in the total derivative,

$$\frac{d\overline{J}_i(\theta; \overline{u}_i'(\theta))}{d\theta} = \frac{\partial \overline{J}_i(\theta; \overline{u}_i)}{\partial \theta}\Big|_{\overline{u}_i = \overline{u}_i'(\theta)} + \frac{\partial \overline{J}_i(\theta; \overline{u}_i)}{\partial \overline{u}_i}\Big|_{\overline{u}_i = \overline{u}_i'(\theta)} \cdot \frac{\partial \overline{u}_i'(\theta)}{\partial \theta} \tag{4}$$

does not vanish, and the often complicated dependence of $\overline{u}_i'(\theta)$ on θ must be explicitly modeled (e.g. by "un-rolling" the iterations of a chosen minimization method yielding $\overline{u}_i'(\theta)$). Otherwise, the estimate for $\nabla_\theta J$ will be biased, and minimization of J will be eventually hindered. Nevertheless, we are interested in such inexact solutions $\overline{u}_i'(\theta)$, that can be obtained in finite time (without warm-starting from a previous estimate), and the question is how close $\overline{u}_i'(\theta)$ has to be to $\overline{u}_i^*(\theta)$ in order to still successfully minimize Eq. 2. Hence, we are interested in algorithms that have the following properties:

1. returns a minimizer (or in general a stationary point) of Eq. 2,
2. does not require storing $\overline{\mathbf{u}} = (\overline{u}_1, \ldots, \overline{u}_N)$ between updates of θ,
3. and is optionally applicable in a stochastic or incremental setting.

We propose two algorithms to minimize Eq. 2, that leverage inexact minimization for the latent variables $\overline{u}_i$ (described in Sections 4 and 5). Our analysis applies to the setting, when each $\overline{J}_i(\theta; \overline{u}_i)$ is convex in $\overline{u}_i$. The basic principle is illustrated in Fig. 1: in iteration t of each of the proposed algorithms, a new upper bound parametrized by $\overline{u}^{(t)}$ is found, that guarantees a sufficient improvement over the previous upper bound according to a respective criterion. This criterion either uses past objective values (Fig. 1(a)) or current gradient information (Fig. 1(b)). In Section 6 we demonstrate the proposed algorithms for large scale robust estimation instances and for training a layered energy-based model.

2 Related Work

Our proposed methods are based on the majorization-minimization (MM) principle [14, 12], which generalizes methods such as expectation-maximization [7, 28, 21] and the convex-concave procedure [33]. A large number of variants and extensions of MM exist. The notion of a (global) majorizer is relaxed in [17, 19], where also a stochastic variant termed MISO (Minimization by Incremental Surrogate Optimization) is proposed. The memory consumption of MISO is $O(ND)$, as sufficient information about each term in Eq. 2 has to be maintained. Here D is the size of the data necessary to represent a surrogate function (i.e. $D = \dim(\overline{u}_i)$). The first-order surrogates introduced in [17] are required to agree with the gradient at the current solution, which is relaxed to asymptotic agreement in [31].

The first of our proposed methods is based on the "generalized MM" method presented in [22], which relaxes the "touching condition" in MM by a looser diminishing gap criterion. Our second method is also a variant of MM, but it is stated such that it easily transfers to a stochastic optimization setting. Since our surrogate functions are only upper bounds of the true objective, the gradient induced by a mini-batch will be biased even at the current solution. This is different from e.g. [37], where noisy surrogate functions are considered, which have unbiased function values and gradients at the current solution. The *stochastic majorization-minimization* [18] and the *stochastic successive upper-bound minimization* (SSUM, [24]) algorithms average information from the surrogate functions gathered during the iterations. Thus, for Lipschitz gradient (quadratic)

surrogates, the memory requirements reduce to $O(D)$ (compared to $O(ND)$ for the original MISO). Several gradient-based methods that are able to cope with noisy gradient oracles are presented in [3, 8, 9] with different assumptions on the objective function and on the gradient oracle,

Majorization-minimization is strongly connected to minimization by alternation (AM). In [6] a "5-point" property is proposed, that is a sufficient condition for AM to converge to a global minimum. Byrne [4] points out that AM (and therefore MM) fall into a larger class of algorithms termed "sequential unconstrained minimization algorithm" (SUMMA).

Contrastive losses such as the one employed in Section 6.2 occur often when model parameters of latent variable models are estimated from training data (e.g. [20, 32]). Such losses can be interpreted either as finite-difference approximations to implicit differentiation [30, 26, 36], as surrogates for the misclassification loss [32], or as approximations to the cross-entropy loss [36]. Thus, contrastive losses are an alternative to the exact gradient computation in bilevel optimization problems (e.g. using the Pineda-Almeida method [23, 2, 25]).

3 Minimization Using Families of Upper Bounds

General setting Let $J : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be a differentiable objective function, that is bounded from below (we choose w.l.o.g. $J(\theta) \geq 0$ for all θ). The task is to determine a minimizer θ^* of J (or stationary point in general).[2] We assume that J is difficult to evaluate directly (e.g. J has the form of Eq. 2), but a differentiable function $\overline{J}(\theta; \overline{u})$ taking an additional argument $\overline{u} \in \mathcal{U} \subseteq \mathbb{R}^D$ is available that has the following properties:

1. $\overline{J}(\theta, \overline{\mathbf{u}}) \geq J(\theta)$ for all $\theta \in \mathbb{R}^d$ and $\overline{\mathbf{u}} \in \mathcal{U}$,
2. $\overline{J}(\theta, \overline{\mathbf{u}})$ is convex in $\overline{\mathbf{u}}$ and satisfies strong duality,
3. $J(\theta) = \min_{\overline{\mathbf{u}} \in \mathcal{U}} \overline{J}(\theta, \overline{\mathbf{u}})$.

This means that $\overline{J}(\theta, \overline{\mathbf{u}})$ is a family of upper bounds of J parametrized by $\overline{\mathbf{u}} \in \mathcal{U}$, and the target objective $J(\theta)$ is given as the lower envelope of $\{\overline{J}(\theta, \overline{\mathbf{u}}) : \overline{\mathbf{u}} \in \mathcal{U}\}$. The second condition implies that optimizing the upper bound for a given θ is relatively easy (but in general it still will require an iterative algorithm). As pointed out in Section 1, $\overline{\mathbf{u}}$ may be very high-dimensional and expensive to maintain in memory. We will absorb the constraint $\overline{u} \in \mathcal{U}$ into $\overline{J}$ and therefore drop this condition in the following.

The baseline algorithm: minimization by alternation The straightforward method to minimize J in Eq. 1/Eq. 2 is by alternating minimization (AM) w.r.t. θ and $\overline{u}$. The downside of AM is, that the entire set of latent variables represented by $\overline{\mathbf{u}}$ has to be stored while updating θ. This can be intractable in machine learning applications when $N \gg 1$ and $D \gg 1$.

[2] By convergence to a stationary point we mean that the gradient converges to 0. Convergence of solution is difficult to obtain in the general non-convex setting.

4 Relaxed Generalized Majorization-Minimization

Our first proposed method extends the generalized majorization-minimization method [22] to the case when computation of J is expensive. Majorization-minimzation (MM, [14, 12]) maintains a sequence of solutions $(\theta^{(t)})_{t=1}^T$ and latent variables $(\overline{\mathbf{u}}^{(t)})_{t=1}^T$ such that

$$\theta^{(t-1)} \leftarrow \arg\min_{\theta} \overline{J}(\theta, \overline{\mathbf{u}}^{(t-1)}) \qquad \overline{u}^{(t)} \leftarrow \arg\min_{\overline{\mathbf{u}}} \overline{J}(\theta^{(t-1)}, \overline{u}). \tag{5}$$

Standard MM requires the following "touching condition" to be satisfied,

$$\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) = J(\theta^{(t-1)}). \tag{6}$$

It should be clear that a standard MM approach is equivalent to the alternating minimization baseline algorithm. In most applications of MM, the domain of the latent variables defining the upper bound is identical to the domain for θ.

Generalized MM relaxes the touching condition to the following one,

$$\begin{aligned}\overline{J}(\theta^{(t-1)},\overline{\mathbf{u}}^{(t)}) &\leq \eta J(\theta^{(t-1)}) + (1-\eta)\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)}) \\ &= \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)}) - \eta\left(\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)}) - J(\theta^{(t-1)})\right),\end{aligned} \tag{7}$$

where $\eta \in (0,1)$ is a user-specified parameter. By construction the gap $d_t := \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)}) - J(\theta^{(t-1)})$ is non-negative. The above condition means that $\overline{\mathbf{u}}^{(t)}$ has to be chosen such that the new objective value $\overline{J}(\theta^{(t)}, \overline{\mathbf{u}}^{(t)})$ is guaranteed to sufficiently improve over the current upper bound $\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)})$,

$$\overline{J}(\theta^{(t)}, \overline{\mathbf{u}}^{(t)}) \leq \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) \leq \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)}) - \eta d_t.$$

It is shown that the sequence $\lim_{t\to\infty} d_t \to 0$, i.e. asymptotically the true cost J is optimized. Since generalized MM decreases the upper bound less aggressively than standard MM, it has an improved empirical ability to reach better local minima in highly non-convex problems [22].

Generalized MM is not directly applicable in our setting, as J is assumed not to be available (or at least expensive to compute, which is exactly we aim to avoid). By leveraging convex duality we have a lower bound for $\underline{J}(\theta, \underline{\mathbf{u}}) \leq J(\theta)$ available. Hence, we modify the generalized MM approach by replacing $J(\theta^{(t-1)})$ with a lower bound $\underline{J}(\theta^{(t-1)}, \underline{\mathbf{u}}^{(t)})$ for a suitable dual parameter $\underline{\mathbf{u}}^{(t)}$, leading to a condition on $\overline{\mathbf{u}}^{(t)}$ and $\underline{\mathbf{u}}^{(t)}$ of the form

$$\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) \leq \eta \underline{J}(\theta^{(t-1)}, \underline{\mathbf{u}}^{(t)}) + (1-\eta)\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)}).$$

This condition still has the significant shortcoming, that both $\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)})$ and $\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)})$ need to be evaluated. While computation of the first quantity is firmly required, evaluation of the second value is unnecessary as we will see in the following. Not needing to compute $\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t-1)})$ also means that the memory associated with $\overline{\mathbf{u}}^{(t-1)}$ can be immediately reused. Our proposed condition on $\overline{\mathbf{u}}^{(t)}$ and $\underline{\mathbf{u}}^{(t)}$ for a *relaxed generalized* MM (or *ReGeMM*) method is given by

$$\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) \leq \eta \underline{J}(\theta^{(t-1)}, \underline{\mathbf{u}}^{(t)}) + (1-\eta)\overline{J}(\theta^{(t-2)}, \overline{\mathbf{u}}^{(t-1)}), \tag{8}$$

Algorithm 1 ReGeMM: Relaxed Generalized Majorization-Minimization

Require: Initial $\theta^{(0)} = \theta^{(-1)}$ and $\overline{\mathbf{u}}^{(0)}$, number of rounds T
for $t = 1, \dots, T$ **do**
  Determine $\overline{\mathbf{u}}^{(t)}$ and $\underline{\mathbf{u}}^{(t)}$ that satisfy Eq. 8
  Set $\theta^{(t)} \leftarrow \arg\min_\theta \overline{J}(\theta, \overline{\mathbf{u}}^{(t)})$
end for
return $\theta^{(T)}$

where $\eta \in (0, 1)$, e.g. $\eta = 1/2$ in our implementation. The resulting algorithm is given in Alg. 1. The existence of a pair $(\overline{\mathbf{u}}^{(t)}, \underline{\mathbf{u}}^{(t)})$ is guaranteed, since both $\underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}^{(t)})$ and $\overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})$ can be made arbitrarily close to $J(\theta^{(t-1)})$ by our assumption of strong duality. We introduce c_t,

$$c_t := \overline{J}(\theta^{(t-2)}, \overline{\mathbf{u}}^{(t-1)}) - \underline{J}(\theta^{(t-1)}, \underline{\mathbf{u}}^{(t)}) \geq 0, \tag{9}$$

and Eq. 8 can therefore be restated as

$$\overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) \leq \overline{J}(\theta^{(t-2)}, \overline{\mathbf{u}}^{(t-1)}) - \eta c_t. \tag{10}$$

Proposition 1. *We have* $\lim_{t\to\infty} c_t = 0$.

Proof. We define $v_t := \overline{J}(\theta^{(t-2)}, \overline{\mathbf{u}}^{(t-1)}) - \eta c_t$. First, observe that

$$\begin{aligned} c_t &= \overline{J}(\theta^{(t-2)}; \overline{\mathbf{u}}^{(t-1)}) - \underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}^{(t)}) \geq \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t-1)}) - \underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}^{(t)}) \\ &\geq \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t-1)}) - J(\theta^{(t-1)}) \geq 0 \end{aligned}$$

(using the relations $\overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t-1)}) \leq \overline{J}(\theta^{(t-2)}; \overline{\mathbf{u}}^{(t-1)})$ and $\underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}) \leq J(\theta^{(t-1)}) \leq \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}})$ for any $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$). We further have

$$\begin{aligned} \sum\nolimits_{t=1}^{T} c_t &= \eta^{-1} \sum\nolimits_{t=1}^{T} \left(\overline{J}(\theta^{(t-2)}; \overline{\mathbf{u}}^{(t-1)}) - v_t \right) \\ &\leq \eta^{-1} \sum\nolimits_{t=1}^{T} \left(\overline{J}(\theta^{(t-2)}; \overline{\mathbf{u}}^{(t-1)}) - \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) \right) \\ &= \eta^{-1} \left(\overline{J}(\theta^{(-1)}; \overline{\mathbf{u}}^{(0)}) - \overline{J}(\theta^{(T-1)}; \overline{\mathbf{u}}^{(T)}) \right) < \infty, \end{aligned}$$

since $\overline{J}$ is bounded from below. In the first line we used the definition of d_t and in the second line we utilized that $\overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) \leq v_t$. The last line follows from the telescopic sum. Overall, we have that

$$\lim_{T\to\infty} \sum\nolimits_{t=1}^{T} c_t = \eta^{-1} \left(\overline{J}(\theta^{(-1)}; \overline{\mathbf{u}}^{(0)}) - \lim_{T\to\infty} \overline{J}(\theta^{(T-1)}; \overline{\mathbf{u}}^{(T)}) \right),$$

which is finite, since J (and therefore $\overline{J}$) is bounded from below. From $c_t \geq 0$ and $\lim_{T\to\infty} \sum_{t=1}^{T} c_t < \infty$ we deduce that $\lim_{T\to\infty} c_t = 0$.

Hence, in analogy with the generalized MM method [22], the upper bound $\overline{J}(\theta^{(t)}, \overline{\mathbf{u}}^{(t)})$ approaches the target objective value $J(\theta^{(t)})$ in the proposed relaxed scheme. This result also implies that finding $\overline{\mathbf{u}}^{(t)}$ will be increasingly harder. This is expected, since one ultimately aims to minimize J. If we additionally assume that the mapping $\theta \mapsto \nabla_\theta \overline{J}(\theta, \overline{\mathbf{u}})$ has Lipschitz gradient for all $\overline{\mathbf{u}}$, then it can be also shown that $\nabla_\theta \overline{J}(\theta^{(t)}, \overline{\mathbf{u}}^{(t)}) \to 0$ (we refer to the supplementary material).

The relaxed generalized MM approach is therefore a well-understood method when applied in a full batch scenario (recall Eq. 2). Since the condition in Eq. 8 is based on all terms in the objectives, it is not clear how it generalizes to an incremental or stochastic setting, when θ is updated using small mini-batches. This is the motivation for developing an alternative criterion to Eq. 8 in the next section, that is based on "local" quantities.

Using constant memory Naive implementations of Alg. 1 require $O(N)$ memory to store $\overline{\mathbf{u}} = (\overline{u}_1, \ldots, \overline{u}_N)$. In many applications the number of terms N is large, but the latent variables $(\overline{u}_i)_i$ have the same structure for all i (e.g. $\overline{u}_i$ represent pixel-level predictions for training images of the same dimensions). If we use a gradient method to update θ, then the required quantities can be accumulated in-place, as shown in the supplementary material. The constant memory algorithm is not limited to first order methods for θ, but any method that accumulates the information needed to determine $\theta^{(t)}$ from $\theta^{(t-1)}$ in-place is feasible (such as the Newton or the Gauss-Newton method).

5 Sufficient Descent Majorization-Minimization

The ReGeMM method proposed above has two disadvantages: (i) the underlying condition is somewhat technical and it is also a global condition, and (ii) the resulting algorithm does not straightforwardly generalize to incremental or stochastic methods, that have proven to be far superior compared to full-batch approaches, especially in machine learning scenarios.

In this section we make the additional assumption on $\overline{J}$, that

$$\overline{J}(\theta', \overline{\mathbf{u}}) \leq \overline{J}(\theta, \overline{\mathbf{u}}) + \nabla_\theta \overline{J}(\theta, \overline{\mathbf{u}})^T(\theta' - \theta) + \frac{L}{2}\|\theta' - \theta\|^2, \tag{11}$$

for a constant $L > 0$ and all $\overline{\mathbf{u}}$. This essentially means, that the mapping $\theta \mapsto \overline{J}(\theta, \overline{\mathbf{u}})$ has a Lipschitz gradient with Lipschitz constant L. This assumption is frequent in many gradient-based minimization methods. Note that the minimizer of the r.h.s. in Eq. 11 w.r.t. θ' is given by $\theta' = \theta - \frac{1}{L}\nabla_\theta \overline{J}(\theta, \overline{\mathbf{u}})$. Hence, we focus on gradient-based updates of θ in the following, i.e. $\theta^{(t)}$ is given by

$$\theta^{(t)} = \theta^{(t-1)} - \frac{1}{L}\nabla_\theta \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}). \tag{12}$$

Combining this with Eq. 11 yields

$$\overline{J}(\theta^{(t)}, \overline{\mathbf{u}}^{(t)}) \leq \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) - \frac{1}{2L}\|\nabla_\theta \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)})\|^2,$$

hence the update from $\theta^{(t-1)}$ to $\theta^{(t)}$ yields a guaranteed reduction of $\overline{J}(\cdot, \overline{\mathbf{u}}^{(t)})$ in terms of the respective gradient magnitude.

We therefore propose the following condition on $(\overline{\mathbf{u}}^{(t)}, \underline{\mathbf{u}}^{(t)})$ based on the current iterate $\theta^{(t-1)}$: for a $\rho \in (0, 1)$ (which is set to $\rho = 1/2$ in our implementation) determine $\overline{\mathbf{u}}^{(t)}$ and $\underline{\mathbf{u}}^{(t)}$ such that

$$\overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) - \underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}^{(t)}) \leq \frac{\rho}{2L} \|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\|^2 \tag{13}$$

This condition requires intuitively, that the duality gap $\overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) - \underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}^{(t)})$ is sufficiently smaller than the reduction of $\overline{J}(\cdot; \overline{u}^{(t)})$ guaranteed by a gradient descent step. Convexity and strong duality of $\overline{J}(\theta; \cdot)$ for each θ allows to determine such a pair $(\overline{\mathbf{u}}^{(t)}, \underline{\mathbf{u}}^{(t)})$ using convex optimization methods. Rearranging the above condition (and using that $\theta^{(t)} = \theta^{(t-1)} - \nabla \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)})/L$) yields

$$\begin{aligned}
\overline{J}(\theta^{(t)}; \overline{\mathbf{u}}^{(t)}) &\leq \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) - \frac{1}{2L} \|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\|^2 \\
&\leq \underline{J}(\theta^{(t-1)}; \underline{\mathbf{u}}^{(t)}) - \frac{1-\rho}{2L} \|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\|^2 \\
&\leq J(\theta^{(t-1)}) - \frac{1-\rho}{2L} \|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\|^2,
\end{aligned}$$

i.e. the upper bound at the new solution $\theta^{(t)}$ is sufficiently below the lower bound (and the true function value) at the current solution $\theta^{(t-1)}$. This can be stated compactly,

$$J(\theta^{(t)}) \leq \overline{J}(\theta^{(t)}; \overline{\mathbf{u}}^{(t)}) \leq J(\theta^{(t-1)}) - \frac{1-\rho}{2L} \|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\|^2, \tag{14}$$

and the sequence $(J(\theta^{(t)})_{t=1}^{\infty}$ is therefore non-increasing. Since we are always asking for a sufficient decrease (in analogy with the Armijo condition), we expect convergence to a stationary solution θ^*. This is the case:

Proposition 2. $\lim_{t \to \infty} \nabla_\theta \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) = 0$.

Proof. By rearranging Eq. 14 we have

$$\begin{aligned}
\sum\nolimits_t \|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\|^2 &\leq \frac{2L}{1-\rho} \sum\nolimits_t \left(J(\theta^{(t-1)}) - J(\theta^{(t)}) \right) \\
&= \frac{2L}{1-\rho} \left(J(\theta^{(0)}) - J(\theta^*) \right) < \infty,
\end{aligned}$$

and therefore $\|\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})\| \to 0$, which implies that $\nabla \overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)}) \to 0$.

We summarize the resulting *sufficient descent MM* (or *SuDeMM*) method in Alg. 2. As with the ReGeMM approach, determining $\overline{u}$ is more difficult when closing in on a stationary point (as $\nabla_\theta \overline{J}(\theta^{(t)}, \overline{u}) \to 0$). The gradient step indicated in line 3 in Alg. 2 can be replaced by any update that guarantees sufficient descent. Finally, in analogy with the ReGeMM approach discussed in the previous section, it is straightforward to obtain a constant memory variant of Alg. 2. The stochastic method described below incorporates both immediate memory reduction from $O(N)$ to $O(B)$, where B is the size of the mini-batch, and faster minimization due to the use of mini-batches.

Algorithm 2 SuDeMM: Sufficient-Descent Majorization-Minimization

Require: Initial $\theta^{(0)}$, number of rounds T
1: **for** $t = 1, \ldots, T$ **do**
2: Determine $\overline{\mathbf{u}}^{(t)}$ and $\underline{\mathbf{u}}^{(t)}$ that satisfy Eq. 13
3: Set $\theta^{(t)} \leftarrow \theta^{(t-1)} - \frac{1}{L}\nabla_\theta \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)})$
4: **end for**
5: **return** $\theta^{(T)}$

5.1 Extension to the stochastic setting

In many machine learning applications J will be of the form of Eq. 2 with $N \gg 1$ being the number of training samples. It is well known that in such settings methods levering the full gradient accumulated over all training samples are hugely outperformed by stochastic gradient methods, which operate on a single training sample (i.e. term in Eq. 2) or, alternatively, on a small mini-batch of size B randomly drawn from the range $\{1, \ldots, N\}$.

It is straightforward to extend Alg. 2 to a stochastic setting working on single data points (or mini-batches) by replacing the objective values $\overline{J}(\theta^{(t-1)}; \overline{\mathbf{u}}^{(t)})$, $\underline{J}(\theta^{(t-1)}; \underline{u}^{(t)})$) and the full gradient $\nabla_\theta \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)})$ with the respective mini-batch counter-parts. The resulting algorithm is depicted in Alg. 3 (for mini-batches of size one). Due to the stochastic nature of the gradient estimate $\nabla_\theta \overline{J}_i(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)})$, both the step sizes $\alpha_t > 0$ and the reduction parameter $\rho_t > 0$ are time-dependent and need to satisfy the following conditions,

$$\sum\nolimits_{t=1}^{\infty} \alpha_t = \infty \qquad \sum\nolimits_{t=1}^{\infty} \alpha_t^2 < \infty \qquad \sum\nolimits_{t=1}^{\infty} \rho_t < \infty. \tag{15}$$

The first two conditions on the step sizes $(\alpha_t)_t$ are standard in stochastic gradient methods, and the last condition on the sequence $(\rho_t)_t$ ensures that the added noise by using time-dependent upper bounds $\overline{J}(\cdot, \overline{\mathbf{u}}^{(t)})$ (instead of the time-independent function $J(\cdot)$) has bounded variance. The constraint on ρ_t is therefore stronger than the intuitively necessary condition $\rho_t \overset{t\to\infty}{\to} 0$. We refer to the supplementary material for a detailed discussion. Due to the small size B of a mini-batch, the values of $\overline{\mathbf{u}}_i^{(t)}$ and $\underline{\mathbf{u}}_i^{(t)}$ in the mini-batch can be maintained, and the restarting strategy outlined in Section 4 is not necessary.

6 Applications

6.1 Robust Bundle Adjustment

In this experiment we first demonstrate the applicability of our proposed ReGeMM schemes to a large scale robust fitting task. The aim is to determine whether ReGeMM is also able to avoid poor local minima (in analogy with the k-means experiment in [22]). The hypothesis is, that optimizing the latent variables just enough to meet the ReGeMM condition (Eq. 8) corresponds to a particular

Algorithm 3 Stochastic Sufficient Descent Majorization-Minimization

Require: Initial $\theta^{(0)}$, number of rounds T
1: **for** $t = 1, \ldots, T$ **do**
2: Uniformly sample i from $\{1, \ldots, N\}$
3: Determine $\overline{u}_i$ and $\underline{u}_i$ that satisfy

$$\overline{J}_i(\theta^{(t-1)}, \overline{u}_i) - \underline{J}_i(\theta^{(t-1)}, \underline{u}_i) \leq \frac{\rho_t}{2} \|\nabla_\theta \overline{J}_i(\theta^{(t-1)}, \overline{u}_i)\|^2 \tag{16}$$

4: Set $\theta^{(t)} \leftarrow \theta^{(t-1)} - \alpha_t \nabla_\theta \overline{J}_i(\theta^{(t-1)}, \overline{u}^{(t)})$
5: **end for**
6: **return** $\theta^{(T)}$

variant of graduated optimization, and therefore will (empirically) return better local minima for highly non-convex problems.

Robust bundle adjustment aims to refine the camera poses and 3D point structure to maximize a log-likelihood given image observations and established correspondences. The unknowns are $\theta = (P_1, \ldots, P_n, X_1, \ldots, X_m)$, where $P_k \in \mathbb{R}^6$ refers to the k-th camera pose and $X_j \in \mathbb{R}^3$ is the position of the j-th 3D point. The cost J is given by

$$J(\theta) = \sum\nolimits_i \psi(f_i(\theta) - m_i), \tag{17}$$

where $m_i \in \mathbb{R}^2$ is the i-th image observation and f_i projects the respective 3D point to the image plane of the corresponding camera. ψ is a so called robust kernel, which generally turns J into a highly non-convex objective functions with a large number of local minima. Following [35] an upper bound $\overline{J}$ is given via half-quadratic (HQ) minimization [11],

$$\overline{J}(\theta, \overline{\mathbf{u}}) = \sum\nolimits_i \Big(\frac{\overline{u}_i}{2} \|f_i(\theta) - m_i\|^2 + \kappa(\overline{u}_i) \Big), \tag{18}$$

where $\kappa : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ depends on the choice for ψ, and $\overline{u}_i$ is identified as the (confidence) weight on the i-th observation. The standard MM approach corresponds essentially to the iteratively reweighted least squares method (IRLS), which is prone to yield poor local minima if θ is not well initialized. For given θ the optimal latent variables $\overline{u}_i^*(\theta)$ are given by $\overline{u}_i^*(\theta) = \omega(\|f_i(\theta) - m_i\|)$, where $\omega(\cdot)$ is the weight function associated with the robust kernel ψ. Joint HQ minimization of $\overline{J}$ w.r.t. θ and $\overline{u}$ is suggested and evaluated in [35], which empirically yields significantly better local minima of J than IRLS. We compare this joint-HQ method (as well as IRLS and an explicit graduated method GOM [34]) with our ReGeMM condition (Eq. 8), where the confidence weights $\overline{\mathbf{u}}$ are optimized to meet but not substantially surpass this criterion: a scale parameter $\sigma \geq 1$ is determined such that $\overline{u}_i^{(t)}$ is set to $\omega(\|f_i(\theta^{(t-1)}) - m_i\|/\sigma)$, and $\overline{\mathbf{u}}$ satisfies the ReGeMM condition (Eq. 8) and

$$\eta' J(\theta^{(t-1)}) + (1 - \eta') \overline{J}(\theta^{(t-2)}, \overline{\mathbf{u}}^{(t-1)}) \leq \overline{J}(\theta^{(t-1)}, \overline{\mathbf{u}}^{(t)}) \tag{19}$$

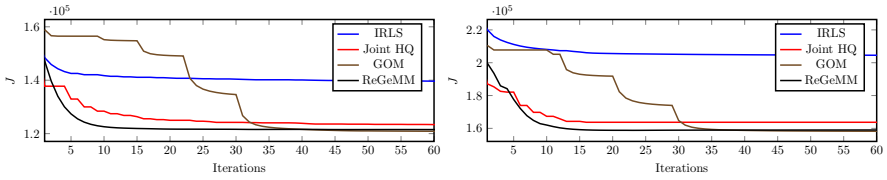


Fig. 2. Objective value w.r.t. number of iterations of a NNLS solver for the Dubrovnik-356 (left) and Venice-427 (right) datasets.

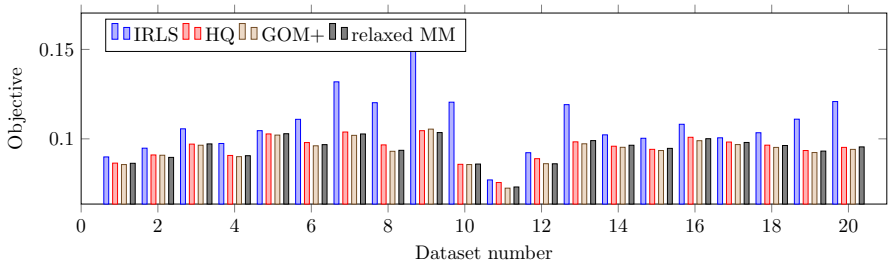


Fig. 3. Final objective values reached by different methods for 20 metric bundle adjustment instances after 100 NNLS solver iterations.

for an $\eta' \in (\eta, 1)$. In our implementation we determine σ using bisection search and choose $\eta' = 3/4$. In this application the evaluation of J is inexpensive, and therefore we use $J(\theta^{(t-1)})$ instead of a lower bound $\underline{J}(\theta^{(t-1)}, \underline{\mathbf{u}}^{(t)})$ in the r.h.s. The model parameters θ are updated for given $\overline{\mathbf{u}}$ using a Levenberg-Marquardt solver. Our choice of ψ is the smooth truncated quadratic cost [35].

In Fig. 2 we depict the evolution of the target objective Eq. 17 for two metric bundle adjustment instances from [1]. The proposed ReGeMM approach (with the initial confidence weights $\overline{\mathbf{u}}$ all set to 1) compares favorably against IRLS, joint HQ [35] and even graduated optimization [34] (that leads only to a slightly better minimum).This observation is supported by comparing the methods using a larger database of 20 problem instances [1] (listed in the supplementary material) in Fig. 3, where the final objective values reached after 100 NNLS iterations by different methods are depicted. ReGeMM is again highly competitive. In terms of run-time, ReGeMM is beetwen 5% and 25% slower per iteration than IRLS in our implementation.

6.2 Contrastive Hebbian Learning

Contrastive Hebbian learning uses an energy model over latent variables to explicitly infer (i.e. minimize over) the network activations (instead of using a predefined rule such as in feed-forward DNNs). Feed-forward DNNs using certain activation functions can be identified as limit case of suitable energy-based models [30, 26, 36]. We use the formulation proposed in [36] due to the underlying convexity of the energy model. In the following we outline that the corresponding

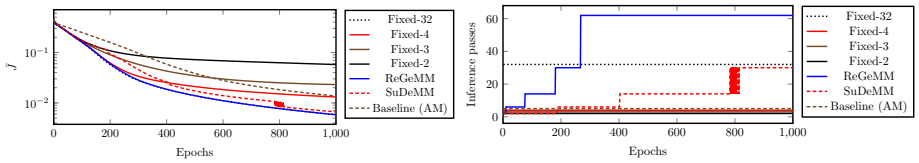


Fig. 4. Objective value w.r.t. number of epochs (left) and the (accumulated) number of inference steps needed to meet the respective criterion (right) in the full-batch setting.

supervised learning task is an instance of Eq. 2. In contrastive Hebbian learning the activations for the network are inferred in two phases: the *clamped phase* uses information from the target label (via a loss function ℓ that is convex in its first argument) to steer the output layer, and the *free phase* does not put any constraint on the output. The input layer is always clamped to the provided training input. The clamped network energy is given by[3]

$$\hat{E}(z;\theta) = \ell(a_L;y) + \frac{1}{2}\left\|z_1 - W_0 x - b_0\right\|^2 + \frac{1}{2}\sum_{k=1}^{L-2}\left\|z_{k+1} - W_k z_k - b_k\right\|^2 \tag{20}$$

subject to $z_k \in \mathcal{C}_k$, where $\mathcal{C}_k$ is a convex set and θ contains all network weights W_k and biases b_k. In order to mimic DNNs with ReLU activations, we choose $\mathcal{C}_k = \mathbb{R}^{n_k}_{\geq 0}$. The loss function is chosen to be the Euclidean loss, $\ell(a_L;y) = \|a_L - y\|^2/2$. The dual network energy can be derived as

$$\hat{E}^*(\lambda;\theta) = -\ell^*(-\lambda_L;y) - \frac{1}{2}\sum\nolimits_{k=1}^{L-1}\left\|\lambda_k\right\|^2 - \lambda_1^T W_0 x + \sum\nolimits_{k=1}^{L}\lambda_k^T b_{k-1} \tag{21}$$

subject to $\lambda_k \geq W_k^T \lambda_{k+1}$ for $k = 1,\ldots,L-1$. If $\ell \equiv 0$, i.e. there is no loss on the final layer output, then we denote the corresponding *free* primal and dual energies by $\check{E}$ and $\check{E}^*$, respectively. Observe that $\hat{E}/\check{E}$ are convex w.r.t. the network activations z, and $\hat{E}^*/\check{E}^*$ are concave w.r.t. the dual variables λ.

Training using contrastive learning Let $\{(x_i, y_i)\}_i$ be a labeled dataset containing N training samples, and the task for the network is to predict y_i from given x_i. The utilized contrastive training loss is given by

$$\begin{aligned} J(\theta) &:= \sum\nolimits_i \left(\min_{\hat{z}} \hat{E}(\hat{z};x_i,y_i,\theta) - \min_{\check{z}} \check{E}(\check{z};x_i,\theta) \right) \\ &= \sum\nolimits_i \min_{\hat{z}} \max_{\check{z}} \left(\hat{E}(\hat{z};x_i,y_i,\theta) - \check{E}(\check{z};x_i,\theta) \right), \end{aligned} \tag{22}$$

which is minimized w.r.t. the network parameters θ. Using duality this saddle-point problem can be restated as pure minimization and maximization tasks [36],

$$\overline{J}(\theta, (\hat{z}_i, \check{\lambda}_i)_{i=1}^N) = \sum\nolimits_i \left(\hat{E}(\hat{z}_i;x_i,y_i,\theta) - \check{E}^*(\check{\lambda}_i;x_i,\theta) \right) \tag{23}$$

[3] We omit the explicit feedback parameter used in [30, 36], since it can be absorbed into the activations and network weights.

and

$$\underline{J}(\theta, (\hat{\lambda}_i, \check{z}_i)_{i=1}^N) = \sum\nolimits_i \left(\hat{E}^*(\hat{\lambda}_i; x_i, y_i, \theta) - \check{E}(\check{z}_i; x_i, \theta) \right). \tag{24}$$

Thus, the latent variables $\overline{u}_i = (\hat{z}_i, \check{\lambda}_i)$ and $\underline{u}_i = (\hat{\lambda}_i, \check{z}_i)$ correspond to primal-dual pairs representing the network activations, and therefore the entire set of latent variables $\overline{\mathbf{u}}$ is very high-dimensional. In this scenario the true cost J is not accessible, since it requires solving a inner minimization problem w.r.t. $\overline{u}_i$ not having a closed form solution (it requires solving a convex QP). Inference (minimization) w.r.t. $\overline{u}_i$ is conducted by coordinate descent, which is guaranteed to converge to a global solution as both $\hat{E}$ and $-\check{E}^*$ are strongly convex [16, 27].

Full batch methods In Fig. 4 we illustrate the evolution of J on a subset of MNIST [15] using a fully connected 784-64($\times$4)-10 architecture for 4 methods: (i) inferring $\overline{u}_i$ with a fixed number of 2, 3, 4 and 32 passes of an iterative method, respectively, (ii) using the ReGeMM condition Eq. 8, and (iii) using the SuDeMM criterion Eq. 13. Inference for $\overline{u}_i$ is continued until the respective criterion is met. Both ReGeMM and SuDeMM use the respective constant memory variants. In this scenario 32 passes are considered sufficient to perform inference, and the ReGeMM and SuDeMM methods track the best curve well. We chose to use the number of epochs (i.e. the number of updates of θ) on the x-axis to align the curves. Clearly, using a fixed number of 2 passes is significantly faster than using 32 or an adaptive but growing number of inference steps. Interestingly, the necessary inference steps grow much quicker (to the allowed maximum of 40 passes) for the ReGeMM condition compared to the SuDeMM test. The baseline method alternates between gradient updates w.r.t. $\overline{\mathbf{u}}$ (using line search) and θ. In all methods the gradient update for θ uses the same fixed learning rate.

Stochastic methods For the stochastic method in Alg. 3 we illustrate the evolution of the objectives values $\overline{J}$ and the number of inference passes in Fig. 5. For MNIST and its drop-in replacements Fashion-MNIST [29] and KMNIST [5] we again use the same 784-64($\times$4)-10 architecture as above. For a greyscale version of CIFAR-10 [13] we employ a 1024-128($\times$3)-10 network. The batch size is 10, and a constant step size is employed. The overall conclusions from Fig. 5 are as follows: using an insufficient number of inference passes yields poor surrogates for the true objective J and it can lead to numerical instabilities due to the biasedness of the gradient estimates. Further, the proposed SuDeMM algorithm yields the lowest estimates for the true objective by gradually adapting the necessary inference precision.[4]

7 Conclusion

We present two approaches to optimize problems with a latent variable structure. Our formally justified methods (i) enable inexact (or truncated) minimization

[4] We refer to the supplementary material for results corresponding to sparse coding as the underlying network model.

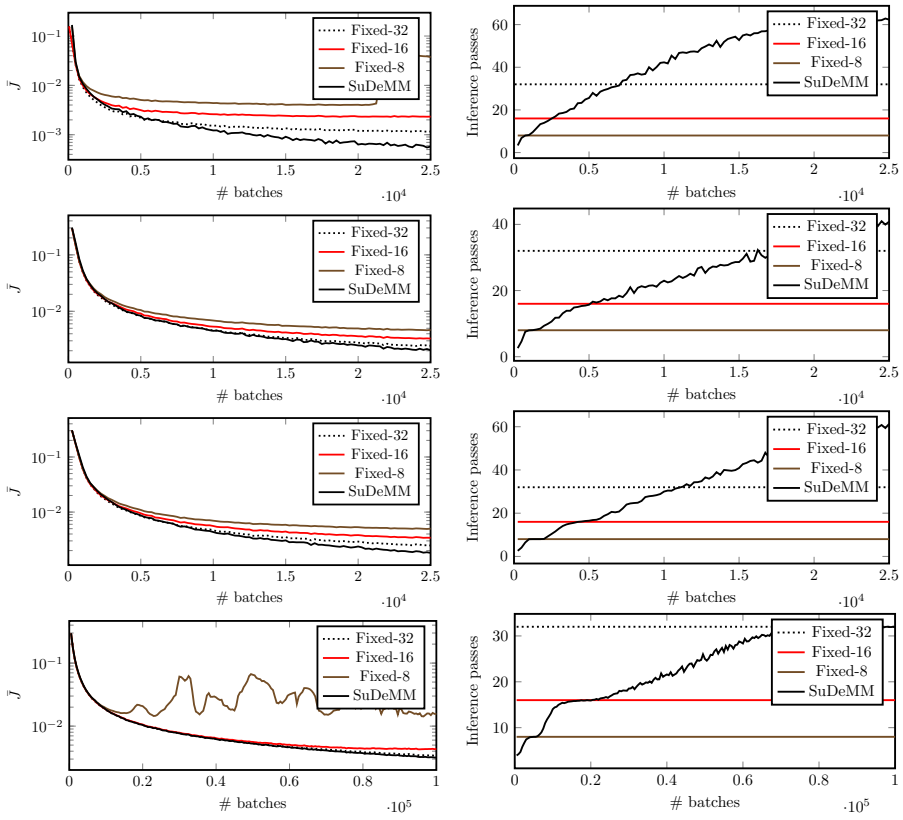


Fig. 5. Objective value w.r.t. number of processed mini-batches (left column) and the number of inference steps needed to meet the respective criterion (right column) for MNIST (1st row), Fashion-MNIST (2nd row), KMNIST (3rd row) and CIFAR-10 (bottom row) using the stochastic gradient method.

over the latent variables and (ii) allow to discard the latent variables between updates of the main parameters of interest. Hence, the proposed methods significantly reduce the memory consumption, and automatically adjust the necessary precision for latent variable inference. One of the two presented methods can be adapted to return competitive solutions for highly non-convex problems such as large-scale robust estimation, and the second method can be run in a stochastic optimization setting in order to address machine learning tasks.

In the future we plan to better understand how turning the proposed ReGeMM inequality condition essentially into an equality constraint can help with solving highly non-convex optimization problems. Further, the presented SuDeMM method enables us to better explore a variety of convex energy-based models in the future.

References

1. Agarwal, S., Snavely, N., Seitz, S.M., Szeliski, R.: Bundle adjustment in the large. In: Proc. ECCV, pp. 29–42. Springer (2010)
2. Almeida, L.B.: A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In: Artificial neural networks: concept learning, pp. 102–111 (1990)
3. Bertsekas, D.P., Tsitsiklis, J.N.: Gradient convergence in gradient methods with errors. SIAM Journal on Optimization **10**(3), 627–642 (2000)
4. Byrne, C.L.: Alternating minimization as sequential unconstrained minimization: a survey. Journal of Optimization Theory and Applications **156**(3), 554–566 (2013)
5. Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical japanese literature. arXiv preprint arXiv:1812.01718 (2018)
6. Csiszár, I., Tusnády, G.E.: Information geometry and alternating minimization procedures. In: Statistics and Decisions (1984)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)
8. Devolder, O., Glineur, F., Nesterov, Y.: First-order methods of smooth convex optimization with inexact oracle. Mathematical Programming **146**(1-2), 37–75 (2014)
9. Dvurechensky, P., Gasnikov, A.: Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. Journal of Optimization Theory and Applications **171**(1), 121–145 (2016)
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32**(9), 1627–1645 (2009)
11. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. **14**(3), 367–383 (1992)
12. Hunter, D.R., Lange, K.: A tutorial on MM algorithms. The American Statistician **58**(1), 30–37 (2004)
13. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
14. Lange, K., Hunter, D.R., Yang, I.: Optimization transfer using surrogate objective functions. Journal of computational and graphical statistics **9**(1), 1–20 (2000)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
16. Luo, Z.Q., Tseng, P.: On the convergence of the coordinate descent method for convex differentiable minimization. Journal of Optimization Theory and Applications **72**(1), 7–35 (1992)
17. Mairal, J.: Optimization with first-order surrogate functions. In: International Conference on Machine Learning. pp. 783–791 (2013)
18. Mairal, J.: Stochastic majorization-minimization algorithms for large-scale optimization. In: Advances in Neural Information Processing Systems. pp. 2283–2291 (2013)
19. Mairal, J.: Incremental majorization-minimization optimization with application to large-scale machine learning. SIAM Journal on Optimization **25**(2), 829–855 (2015)
20. Movellan, J.R.: Contrastive Hebbian learning in the continuous hopfield model. In: Connectionist Models, pp. 10–17. Elsevier (1991)

21. Neal, R.M., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in graphical models, pp. 355–368. Springer (1998)
22. Parizi, S.N., He, K., Aghajani, R., Sclaroff, S., Felzenszwalb, P.: Generalized majorization-minimization. In: International Conference on Machine Learning. pp. 5022–5031 (2019)
23. Pineda, F.J.: Generalization of back-propagation to recurrent neural networks. Physical review letters **59**(19), 2229 (1987)
24. Razaviyayn, M., Sanjabi, M., Luo, Z.Q.: A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks. Mathematical Programming **157**(2), 515–545 (2016)
25. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2008)
26. Scellier, B., Bengio, Y.: Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. Frontiers in computational neuroscience **11**, 24 (2017)
27. Wright, S.J.: Coordinate descent algorithms. Mathematical Programming **151**(1), 3–34 (2015)
28. Wu, C.J.: On the convergence properties of the EM algorithm. The Annals of statistics pp. 95–103 (1983)
29. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
30. Xie, X., Seung, H.S.: Equivalence of backpropagation and contrastive Hebbian learning in a layered network. Neural computation **15**(2), 441–454 (2003)
31. Xu, C., Lin, Z., Zhao, Z., Zha, H.: Relaxed majorization-minimization for non-smooth and non-convex optimization. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
32. Yu, C.N.J., Joachims, T.: Learning structural SVMs with latent variables. In: Proceedings of the 26th annual international conference on machine learning. pp. 1169–1176 (2009)
33. Yuille, A.L., Rangarajan, A.: The concave-convex procedure. Neural computation **15**(4), 915–936 (2003)
34. Zach, C., Bourmaud, G.: Descending, lifting or smoothing: Secrets of robust cost optimization. In: Proc. ECCV (2018)
35. Zach, C.: Robust bundle adjustment revisited. In: Proc. ECCV. pp. 772–787. Springer International Publishing (2014)
36. Zach, C., Estellers, V.: Contrastive learning for lifted networks. In: British Machine Vision Conference (2019)
37. Zhang, H., Zhou, P., Yang, Y., Feng, J.: Generalized majorization-minimization for non-convex optimization. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. pp. 4257–4263. AAAI Press (2019)