Majorization-Minimization for Manifold Embedding

Zhirong Yang[1,2,3], **Jaakko Peltonen**[1,3,4] **and Samuel Kaski**[1,2,3]
[1]Helsinki Institute for Information Technology HIIT, [2]University of Helsinki,
[3]Aalto University, [4]University of Tampere

Abstract

Nonlinear dimensionality reduction by manifold embedding has become a popular and powerful approach both for visualization and as preprocessing for predictive tasks, but more efficient optimization algorithms are still crucially needed. Majorization-Minimization (MM) is a promising approach that monotonically decreases the cost function, but it remains unknown how to tightly majorize the manifold embedding objective functions such that the resulting MM algorithms are efficient and robust. We propose a new MM procedure that yields fast MM algorithms for a wide variety of manifold embedding problems. In our majorization step, two parts of the cost function are respectively upper bounded by quadratic and Lipschitz surrogates, and the resulting upper bound can be minimized in closed form. For cost functions amenable to such QL-majorization, the MM yields monotonic improvement and is efficient: In experiments, the newly developed MM algorithms outperformed five state-of-the-art optimization approaches in manifold embedding tasks.

1 Introduction

Nonlinear dimensionality reduction (NLDR) is crucial for visualization of data during the first steps of exploratory data analysis and can also be helpful as preprocessing for machine learning and data mining tasks. Several NLDR approaches are based on manifold embedding, which aims to discover an underlying lower-dimensional manifold of the data embedded in the high-dimensional feature space, and then unfold the manifold. Manifold embedding research has led to applications for example in computer vision, social computing, bioinformatics, and natural language processing. Many manifold embedding methods have been introduced, including methods based on eigendecompositions such as Principal Component Analysis (PCA), Isomap [27], and Locally Linear Embedding (LLE) [25], and recent methods including Stochastic Neighbor Embedding [10, 30], Elastic Embedding [4], and the Neighbor Retrieval Visualizer [32].

Most recent well-performing manifold embedding approaches involve unconstrained minimization of a non-convex cost function. However, algorithms proposed so far, both dedicated approaches and those based on conventional iterative optimization, have at least some of the following drawbacks: (i) they cannot guarantee that the cost is decreased after each iteration without resorting to line search strategies; (ii) they often converge slowly; and (iii) they are sensitive to initializations. Majorization-Minimization (MM, [23]) algorithms can overcome the first drawback, but their performance depends on construction of an upper bound. For complicated cost functions, including manifold embedding cost functions, it remains unknown how to design a tight upper-bounding function that leads to an efficient and robust MM algorithm.

We propose a new approach for MM algorithms using a new upper-bounding technique QL-majorization. In QL-majorization, partial Hessian information is encoded in a quadratic surrogate for a part of the cost function, and the remaining part is majorized by a Lipschitz surrogate. The MM algorithms produced by QL-majorization have notable advantages. The produced majorization function can be minimized in a closed form without line searches. The resulting algorithms (i) monotonically decrease the manifold embedding cost, (ii) often converge faster than other optimizers, and (iii) are less sensitive to the starting point.

The resulting MM approach is applicable to all machine learning tasks whose cost functions satisfy simple requirements; as examples we develop MM algorithms

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

for several prominent manifold embedding methods that optimize the embedding based on various information divergences and embedding kernels. In experiments on manifold embedding problems, the MM algorithms produced by QL-majorization converge faster and yield better-quality embeddings than state-of-the-art optimizers. The MM algorithms improve performance consistently, regardless of whether the compared methods start from simple random initializations or with a state-of-the-art initialization strategy.

2 Manifold Embedding

Nonlinear dimensionality reduction (NLDR) finds a low-dimensional mapping $Y = [y_1, \ldots, y_N]^T \in \mathbb{R}^{N\times r}$ of N high-dimensional data objects. In our experiments we focus on $r = 2$ dimensional outputs as is common in visualization tasks, but the methods are applicable to other r as well. NLDR methods based on finding and unfolding a lower-dimensional manifold of data can be called manifold embedding methods. Successful manifold embedding methods include Isomap [27], Locally Linear Embedding [25], Laplacian Eigenmap [1], Maximum Variance Unfolding [34], Stochastic Neighbor Embedding [10, 30], Elastic Embedding [4], Neighbor Retrieval Visualizer [32], and several others; see [31] for a recent review. Many modern manifold embedding methods are based on a nonnegative square matrix P that encodes the pairwise proximities between the high-dimensional data objects. Correspondingly, let Q denote the matrix of proximities between the lower-dimensional mapped points. The manifold embedding problem can be formulated as minimization of a cost function $\mathcal{J}(Y)$ over Y. The cost function is often non-convex and typically depends on P and Q where Q is computed from Y.

Typical non-convex optimization methods in manifold embedding involve update steps where constant step sizes do not guarantee decrease of the objective; expensive line search of the step size is often needed to decrease $\mathcal{J}$. In contrast, algorithms such as expectation-maximization (EM) would be guaranteed to monotonically decrease the cost in each update. EM algorithms are based on minimizing an auxiliary bound of a cost, and are theoretically appealing as each step analytically updates parameters to yield the global optimum of the bound. It would be appealing to use the same technique for manifold embedding. However, unlike in standard applications of EM, for many manifold embedding cost functions it is difficult to find an auxiliary bounding that would yield efficient and robust optimization; one reason is that the costs involve complicated interactions between objects in the mapped space, thus the costs are complicated functions of Y.

We now propose a new technique to build upper bounds; this makes it possible to apply an EM-type optimization technique in manifold embedding. Below we use Y, $\widetilde{Y}$, and Y^{new} to respectively denote the current estimate, the variable, and the new estimate. Similarly, the proximities $\widetilde{Q}$ are computed from the variable $\widetilde{Y}$ and the Q from the current estimate. In this paper, matrix-wise summation is off-diagonal, i.e., $\sum_{ij}$ means $\sum_{ij:i\neq j}$.

3 QL-Upper Bounds for MM

Majorization-Minimization (MM) is an iterative optimization method, of which EM is a special case. The MM iterations guarantee decrease of the cost function after each iteration. To minimize $\mathcal{J}(\widetilde{Y})$ with respect to $\widetilde{Y}$, MM first constructs an auxiliary function $G(\widetilde{Y}, Y)$ such that $G(\widetilde{Y}, Y) \geq \mathcal{J}(\widetilde{Y})$ and $G(Y, Y) = \mathcal{J}(Y)$ for all Y (this construction is called majorization), and next solves $Y^{\text{new}} = \arg\min_{\widetilde{Y}} G(\widetilde{Y}, Y)$ (this step is then called minimization). Iterating these two steps thus monotonically reduces $\mathcal{J}$ because $\mathcal{J}(Y) = G(Y, Y) \geq G(Y^{\text{new}}, Y) \geq \mathcal{J}(Y^{\text{new}})$.

The key to a good MM algorithm is to construct $G(\widetilde{Y}, Y)$ so that it can be analytically minimized with respect to $\widetilde{Y}$ but is still close to the cost $\mathcal{J}(\widetilde{Y})$. Finding the auxiliary function $G(\widetilde{Y}, Y)$ depends on the structure of $\mathcal{J}(\widetilde{Y})$. Below we propose a two-phase procedure, called QL-majorization, to develop such an upper-bounding function for manifold embedding.

QL-majorization. Assume the cost function can be divided into two parts:

$$\mathcal{J}(\widetilde{Y}) = A(P, \widetilde{Q}) + B(P, \widetilde{Q}), \tag{1}$$

where A satisfies a simple upper bound condition described below. We can then construct the auxiliary function G in two phases which we call *quadratification* and *Lipschitzation*.

- (*Quadratification*) Let the function A have a simple upper bound consisting of pairwise-quadratic terms of the form $A(P, \widetilde{Q}) \leq \sum_{ij} W_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2 + \text{constant}$, where the multipliers W_{ij} do not depend on $\widetilde{Y}$. Then $\mathcal{J}(\widetilde{Y}) \leq \mathcal{H}(\widetilde{Y}, Y)$ where

$$\mathcal{H}(\widetilde{Y}, Y) = \sum_{ij} W_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2 + B(P, \widetilde{Q}) + \text{const.} \tag{2}$$

  For tight majorization, we also require $\mathcal{H}$ to share the tangent with $\mathcal{J}$ at the current estimate, i.e. $\mathcal{H}(Y, Y) = \mathcal{J}(Y)$ and $\frac{\partial \mathcal{H}}{\partial \widetilde{Y}}\Big|_{\widetilde{Y}=Y} = \frac{\partial \mathcal{J}}{\partial \widetilde{Y}}\Big|_{\widetilde{Y}=Y}$. Finding $\mathcal{H}$ is often accomplished by using the concavity/convexity of and within $\mathcal{J}$; we show examples in Sections 4 and 5.

- (*Lipschitzation*) Upper bound B by its Lipschitz surrogate such that $\mathcal{H}(\widetilde{Y}, Y) \leq G(\widetilde{Y}, Y)$, where

$$G(\widetilde{Y}, Y) = \sum_{ij} W_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2 + \left\langle \Psi, \widetilde{Y} - Y \right\rangle + \frac{\rho}{2}\|\widetilde{Y} - Y\|^2 + \text{const.}, \quad (3)$$

  with $\Psi = \frac{\partial B}{\partial \widetilde{Y}}\Big|_{\widetilde{Y}=Y}$ and ρ is the Lipschitz constant of $B(P, \widetilde{Q})$.

The resulting upper bound $G(\widetilde{Y}, Y)$ can be minimized with respect to $\widetilde{Y}$ in closed form by setting the gradient of $G(\widetilde{Y}, Y)$ with respect to $\widetilde{Y}$ to zero and solving for $\widetilde{Y}$, which gives the update rule

$$Y^{\text{new}} = \left(2\mathcal{L}_{W+W^\top} + \rho I\right)^{-1}\left(-\Psi + \rho Y\right), \quad (4)$$

where $\mathcal{L}_M$ denotes the Laplacian of the subscripted matrix M, that is, $\mathcal{L}_M = \Lambda - M$ where Λ is diagonal and $\Lambda_{ii} = \sum_j M_{ij}$. For some functions B the Lipschitz constant ρ is easy to compute; otherwise, if ρ is unknown, it is sufficient to use simple backtracking [20] to enforce point-wise majorization $G(Y^{\text{new}}, Y) \geq \mathcal{J}(Y^{\text{new}})$. The resulting method is described in Algorithm 1.

By the above construction, we achieve the following guarantees of monotonicity (and thus objective convergence if $\mathcal{J}$ is lower-bounded) and stationary points (proofs in the supplemental document):

Theorem 1. *$\mathcal{J}(Y^{new}) \leq \mathcal{J}(Y)$ after applying Eq. 4.*

Theorem 2. *If $\mathcal{J}$ is a bounded function, iteratively applying Eq. 4 converges to a stationary point of $\mathcal{J}$.*

Note that similar to EM, we do not claim convergence to local optima for the proposed MM updates.

The computational cost of the proposed MM algorithm is $O(|E| + N\log N)$, where $|E|$ is the number of nonzeros in P and N is the number of data points. Algorithm 1 involves an outer loop to iteratively update Y and an inner loop to search for ρ. In each outer loop iteration, the algorithm calculates W, $\mathcal{L}_{W+W^T}$, and Ψ, where the computational cost is $O(|E| + N\log N)$ by using speedup trees (see e.g. [29, 36]). Calculating $G(Y^{\text{try}}, Y)$ in the inner loop is cheap with W and Ψ already computed. Thus the cost of verifying $G(Y^{\text{try}}, Y) \geq \mathcal{J}(Y^{\text{try}})$ in each inner loop iteration is dominated by the evaluation of $\mathcal{J}(Y^{\text{try}})$, which is also $O(|E| + N\log N)$. Lastly, applying Eq. 4 requires solving a linear system, which can be done by a few conjugate gradient steps, each with $O(|E|)$ cost.

Algorithm 1 QL-Majorization-Minimization algorithm with backtracking for Manifold Embedding

Input: proximity matrix P, initial $\rho > 0$, inflating factor $\nu > 1$, and initial Y.
repeat
  $\rho \leftarrow \rho/\nu$; calculate W, $\mathcal{L}_W$, and Ψ
  while TRUE **do**
    Apply Eq. 4 to get Y^{try}
    if $G(Y^{\text{try}}, Y) \geq \mathcal{J}(Y^{\text{try}})$ **then**
      break;
    end if
    $\rho \leftarrow \rho \times \nu$
  end while
  $Y \leftarrow Y^{\text{try}}$
until stopping criterion is met
Output: low-dimensional representations Y.

4 Example: MM for t-SNE

We use the method of Section 3 to introduce an MM algorithm for t-Distributed Stochastic Neighbor Embedding (t-SNE, [30]). Given $\sum_{ij} P_{ij} = 1$, the t-SNE cost function can be divided as $\mathcal{J}(\widetilde{Y}) = \sum_{ij} P_{ij} \ln \frac{P_{ij}}{\widetilde{Q}_{ij}} = A(P, \widetilde{Q}) + B(\widetilde{Q}) + \text{constant}$, where $\widetilde{Q}_{ij} = \frac{(1+\|\tilde{y}_i - \tilde{y}_j\|^2)^{-1}}{\sum_{ab}(1+\|\tilde{y}_a - \tilde{y}_b\|^2)^{-1}}$, $A(P, \widetilde{Q}) = \sum_{ij} P_{ij}\ln(1 + \|\tilde{y}_i - \tilde{y}_j\|^2)$, and $B(\widetilde{Q}) = \ln\sum_{ij}(1 + \|\tilde{y}_i - \tilde{y}_j\|^2)^{-1}$. The concave function ln() can be majorized by its tangent line. Thus we have $A(P, \widetilde{Q}) \leq \sum_{ij} P_{ij}\frac{\|\tilde{y}_i - \tilde{y}_j\|^2}{1+\|y_i - y_j\|^2} + \text{constant}$, i.e. $W_{ij} = P_{ij}q_{ij}$ with $q_{ij} = (1 + \|y_i - y_j\|^2)^{-1}$ in quadratification. Therefore a majorization function of t-SNE is $G(\widetilde{Y}, Y) = \sum_{ij} P_{ij}q_{ij}\|\tilde{y}_i - \tilde{y}_j\|^2 + \left\langle \frac{\partial B}{\partial \widetilde{Y}}\Big|_{\widetilde{Y}=Y}, \widetilde{Y} - Y \right\rangle + \frac{\rho}{2}\|\widetilde{Y} - Y\|^2 + \text{constant}$. Zeroing the gradient of $G(\widetilde{Y}, Y)$ to $\widetilde{Y}$ gives the MM update rule for t-SNE:

$$Y^{\text{new}} = \left(\mathcal{L}_{P\circ q} + \frac{\rho}{4}I\right)^{-1}\left(\mathcal{L}_{Q\circ q}Y + \frac{\rho}{4}Y\right), \quad (5)$$

where $\mathcal{L}_M$ again denotes the Laplacian of the subscripted matrix M, and $\circ$ denotes elementwise product. This update rule implements Eq. 4 for t-SNE and is used in Algorithm 1 to yield the MM algorithm for t-SNE. Because $B(\widetilde{Q})$ is smooth and has a finite upper bound of its derivative, its Lipschitz constant is also finite (i.e. the finite upper-bound of ρ).

We provide more examples in the supplemental document, where we develop MM algorithms for Elastic Embedding [4], Stochastic Neighbor Embedding (SNE) with Gaussian kernel [10], symmetric SNE (s-SNE; [30]), Neighbor Retrieval Visualizer (NeRV; [32]), LinLog [21], and Multidimensional Scaling with kernel-strain (MDS-KS; [3]).

5 MM for Neighbor Embedding

The MM development procedure can be easily applied not only to t-SNE but to a large class of problems including many manifold embedding problems. The quadratification step is easy for all cost functions having a similar general form as defined below (proof in the supplemental document).

Theorem 3. *If (i)* $\mathcal{J}(\widetilde{Y}) = \sum_{ij} A_{ij}(P_{ij}, \widetilde{Q}_{ij}) + B(P, \widetilde{Q})$*, that is,* $A(P,Q)$ *in Eq. 1 is additively separable, and (ii)* A_{ij} *is concave in* $\|\tilde{y}_i - \tilde{y}_j\|^2$*, then in Eq. 4,* $W_{ij} = \frac{\partial A_{ij}}{\partial \|\tilde{y}_i - \tilde{y}_j\|^2}\Big|_{\widetilde{Y}=Y}$.

Next we show that a large collection of manifold embedding cost functions, especially those of Neighbor Embedding (NE, [37]) methods, fulfill the conditions of Theorem 3, and thus the above quadratification inserted in Eq. 4 and Algorithm 1 yields an MM algorithm for them. NE minimizes the discrepancy between the input proximities P and the output proximities $\widetilde{Q}$ over $\widetilde{Y}$. The discrepancy is measured by an information divergence $D(P||\widetilde{Q})$, which in this work is from the α-, β-, γ- and Rényi-divergence families. To be concrete, we parameterize [35] $\widetilde{Q}_{ij} = \big(c + a\|\tilde{y}_i - \tilde{y}_j\|^2\big)^{-b/a}$ for $a \geq 0$, $b > 0$, and $c \geq 0$, which includes many existing manifold embedding cost functions (shown in the supplemental document; note that the limit $a \to 0$ yields Gaussian proximities).

Theorem 4. *Suppose* D *is one of the* α- *or* β-*divergences. Then the cost function* $\mathcal{J}(\widetilde{Y}) = D(P||\widetilde{Q})$ *fulfills both conditions in Theorem 3 when* $\alpha \in (0, 1 + a/b]$ *or* $\beta \in [1 - a/b, \infty)$.

For the Cauchy kernel ($a = b = c = 1$), the above ranges include the most commonly used divergences such as (non-normalized) Kullback-Leibler (KL) ($\alpha \to 1$), dual Kullback-Leibler ($\alpha \to 0$), Hellinger ($\alpha = 1/2$), χ^2 ($\alpha = 2$), Itakura-Saito ($\beta \to 0$), and squared Euclidean ($\beta = 2$).

For nonseparable divergences, such as the normalized KL, we can convert the divergences to their separable counterparts by the following optimization equivalence with an additional scaling factor λ:

Theorem 5. *(from [37]) Let* D_α*,* D_β*,* D_γ*, and* D_r *denote the* α-*,* β-*,* γ- *and Rényi-divergences. Then*

$$\arg\min_{\widetilde{Y}} D_{\gamma\to\tau}(P||\widetilde{Q}) = \arg\min_{\widetilde{Y}} \big[\min_{\lambda\geq 0} D_{\beta\to\tau}(P||\lambda\widetilde{Q})\big],$$

$$\arg\min_{\widetilde{Y}} D_{r\to\tau}(P||\widetilde{Q}) = \arg\min_{\widetilde{Y}} \big[\min_{\lambda\geq 0} D_{\alpha\to\tau}(P||\lambda\widetilde{Q})\big].$$

The optimization for nonseparable divergences thus proceeds by interleaving minimizations over λ, which is given in closed form, with minimizations over $\widetilde{Y}$ given by the proposed MM algorithm.

6 Related Work

Majorization-Minimization algorithms date back to 1970's. Ortega and Rheinboldt enunciated the principle in the context of line search methods [23]. Later Expectation-Maximization as an important special case of MM was proposed by Dempster et al. [6]. In the same year, de Leeuw and Heiser presented an MM algorithm for multidimensional scaling [5]. There are many subsequent appearances of MM in various applications, for example, robust regression [11], quadratic bounding principle [2], medical imaging (e.g. [16, 24]), quantile regression [13], survival analysis [14], paired and multiple comparisons [12], variable selection [15], DNA sequence analysis [26], discriminant analysis [18], IRLS (e.g. [38]), image restoration [7] and hyperparameter learning [8]. Recently Mairal also proposed a set of stochastic MM algorithms for large-scale optimization [19]. The key point in MM development is to devise a tight bounding function which can be efficiently optimized. Generic methods for constructing the bounding function include Jensen's inequality for a convex function, tangent plane (supporting hyper-plane) of a concave function, upper-bounding surrogates due to Lipschitz continuity, quadratic upper bound of the second-order Taylor expansion [2], inequality between the generalized means, and Cauchy-Schwartz inequality. Although MM is familiar in the machine learning literature, it remains unknown how to construct the bounding function for modern manifold embedding (ME) objectives such as t-SNE. These objectives are highly non-convex and cannot be majorized by a single inequality listed above.

The proposed MM algorithm employs two phases of majorization: quadratification and Lipschitzation. Using *both* upper bounds distinguishes our method from other optimization approaches.

Without quadratification, minimizing the Lipschitz surrogate $\mathcal{J}(Y) + \Big\langle \nabla, \widetilde{Y} - Y \Big\rangle + \frac{\rho}{2}\|\widetilde{Y} - Y\|^2$ of the whole cost function $\mathcal{J}(\widetilde{Y})$ leads to the gradient descent (GD) method $Y^{\text{new}} = Y - \frac{1}{\rho}\nabla$, where $\nabla = \frac{\partial \mathcal{J}}{\partial \widetilde{Y}}\big|_{\widetilde{Y}=Y}$. The GD optimization is slow and often falls into poor local optima (see [33] and also Section 7) because it only uses the first-order derivative information. There are methods that approximate the second-order derivatives by using the history of gradients, for example, Limited-memory BFGS (L-BFGS) [22]. However, in Section 7 we show that such approximations are often inaccurate for the manifold embedding problem and thus subject to slow convergence.

On the other hand, without the Lipschitz surrogate it is often difficult to use quadratification alone, that is, to find a quadratic upper bound for the whole cost

Table 1: Mean converged t-SNE cost function values (± standard deviation) obtained with the compared algorithms. We use exact calculation for datasets where $N \leq 20K$, and Barnes-Hut approximation otherwise. The boldface cells show the best mean converged cost function value in each row.

dataname	N	GD	L-BFGS	MOMENTUM	SD	FPHSSNE	MM
SCOTLAND	108	0.96±0.02	0.92±0.02	1.26±0.28	7.27±5.89	1.22±0.14	**0.91±0.01**
COIL20	1.4K	0.98±0.04	0.84±0.03	0.84±0.02	0.80±0.01	1.02±0.04	**0.79±0.00**
7SECTORS	4.6K	2.29±0.02	2.36±0.02	2.18±0.01	2.20±0.18	2.30±0.01	**2.09±0.01**
RCV1	9.6K	3.09±0.11	2.97±0.04	2.60±0.01	2.52±0.01	2.77±0.02	**2.47±0.01**
PENDIGITS	11K	2.31±0.09	2.46±0.10	1.86±0.04	1.78±0.09	2.19±0.03	**1.67±0.01**
MAGIC	19K	3.35±0.09	3.71±0.05	2.80±0.03	2.44±0.02	3.16±0.08	**2.37±0.01**
20NEWS	20K	4.95±0.07	4.24±0.06	3.51±0.03	3.35±0.03	3.95±0.11	**3.22±0.02**
LETTERS	20K	3.03±0.21	2.69±0.10	1.89±0.03	9.06±11.58	2.61±0.10	**1.44±0.01**
SHUTTLE	58K	6.20±0.81	3.20±0.13	2.17±0.02	1.60±0.11	2.91±0.01	**1.47±0.05**
MNIST	70K	6.91±0.83	5.14±0.11	3.83±0.04	3.47±0.02	4.14±0.04	**3.42±0.02**
SEISMIC	99K	8.65±0.00	7.54±1.43	4.32±0.03	3.99±0.15	4.58±0.03	**3.85±0.01**
MINIBOONE	130K	9.14±0.00	9.14±0.00	4.12±0.03	3.55±0.04	4.58±0.02	**3.54±0.02**

$\mathcal{J}(\widetilde{Y})$. Mere quadratification can be used in classical multidimensional scaling, where one can write out the weighted least-square error and majorize the cross term by the Cauchy-Schwarz inequality. This is called the stress majorization strategy [5, 9, 17], which brings a better convergence rate than the steepest descent method, because it uses a more accurate approximation to the Hessian. However, such a majorization strategy cannot, in general, be extended to other manifold embedding objectives, especially for the terms that are not known to be convex or concave. These difficult terms are upper bounded by Lipschitz surrogates in our method, as long as they are smooth in Y.

Spectral Direction (SD) [33] gives a search direction which has a similar form as Eq. 4. Using the same decomposition form [4, 33] as in Eq. 1, SD employs the quasi-Newton strategy where it keeps only the second-order derivatives of A and discards those of B. SD heuristically adds a very small positive ϵ to the diagonal of partial Hessian H (e.g. $\epsilon = 10^{-10} \min_i\{H_{ii}\}$) as a remedy for positive definiteness. A similar search direction was earlier used in [28] with $\epsilon = 0$. According to our derivation, the SD strategy is close to linearizing the B term in Eq. 1 throughout the iterations. Although ρ in our MM algorithm has a similar role as ϵ, we emphasize that ρ can adaptively change in a much broader region, which is critical for efficiency and robustness. Moreover, unlike the previous methods (e.g. [22, 28, 33]), our MM algorithm does not use line search but instead uses a curved exploration trajetory in its inner loop. We will show this is more effective in minimizing the ME cost functions.

7 Experiments

We compare the performance of the proposed MM algorithm with five other optimizers on 12 publicly available datasets from different domains. In most experiments we focus on t-SNE because it is the most used method in the machine learning community. We also present preliminary results of other manifold embedding methods beyond t-SNE.

For vectorial datasets, the input to t-SNE is the k-Nearest Neighbor matrix p (k-NN, k=10 in all reported results), with $p_{ij} = 1$ if j is among the k nearest neighbors of i or vice versa, and $p_{ij} = 0$ otherwise. Conclusions are the same for $k = 15$ and $k = 20$. For undirected network data, we simply use the (weighted) adjacency matrix as p. We then normalize $P_{ij} = p_{ij} / \sum_{ab} p_{ab}$.

The other compared optimizers include gradient descent with line search (GD), Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [22], gradient descent with momentum (MOMENTUM) [30], spectral direction with line search (SD) [33], and a fixed-point algorithm for heavy-tailed s-SNE (FPHSSNE) [35]. We use default settings in all the other algorithms. For MM, we used $\nu = 2$ and initial $\rho = 10^{-6}$ throughout all experiments.

Each compared method was run for each dataset ten times. In each run, all algorithms start from the same random initialization $Y = \text{randn}(N, 2) \times 10^{-4}$. Different random seeds were used across the ten runs. An algorithm stops at the t-th iteration if (i) $t \geq 3000$, (ii) it consumes more than 20 hours, (iii) the relative cost function change $|\mathcal{J}_t - \mathcal{J}_{t-1}|/|\mathcal{J}_{t-1}| < 10^{-4}$, or (iv) $\|Y^{\text{new}} - Y\|_F / \|Y\|_F < 10^{-8}$.

7.1 Small data sets

In the first group of experiments, the compared optimizers were tested on eight data sets with $N \leq 20,000$ data objects. For these small-scale manifold embedding tasks, we can use exact calculation for the objective function and gradient, and an exact linear system solver for Eq. 5.

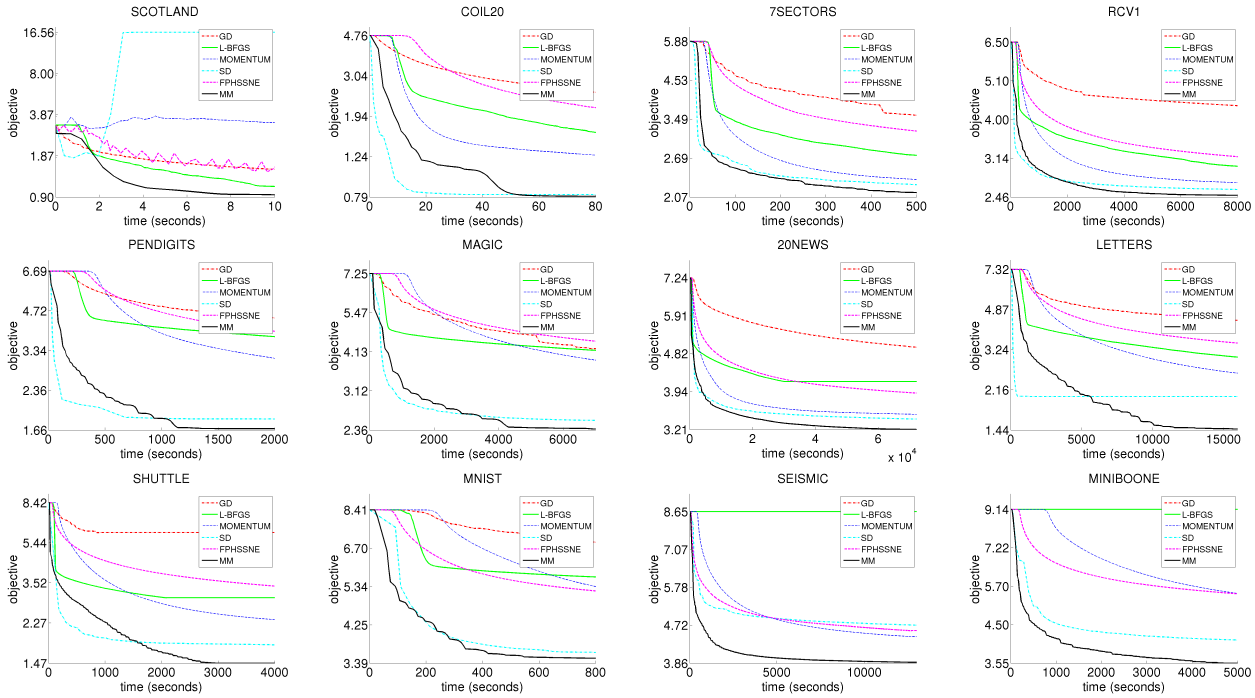

Figure 1: Evolution of the t-SNE objective (cost function value) as a function of time for the compared algorithms. The first and second rows were exactly calculated, while the third row uses Barnes-Hut approximation.

The eight plots in the first two rows of Fig. 1 illustrate the t-SNE objective evolution as a function of time. The results show that MM is clearly faster than GD, L-BFGS, MOMENTUM and FPHSSNE. For some data sets such as `COIL20` and `LETTERS`, although MM is slower than SD at the beginning, it overtakes SD and achieves better cost function values in all plots. For `SCOTLAND`, the SD curve jumps to a useless high value after about three seconds. Note that the MM curves can be coarser than the other methods that employ small steps, because MM usually requires fewer iterations and makes no assumption on the curvature of the objective function.

In Table 1, the top eight rows of results show that MM returns the best mean converged cost function values for all tested small data sets. Moreover, MM also achieves the smallest standard deviations, which indicates the MM with QL-majorization is less sensitive to different random starting layouts. In contrast, the other optimizers can be much more sensitive for certain data sets (e.g., SD for `SCOTLAND` and `LETTERS`).

The resulting visualizations of `20NEWS` are shown in Fig. 2. We can see that the layout learned by using MM is the best in terms of the smallest t-SNE cost function value and of identifying the 20 newsgroups. We quantify the latter performance by the area under the precision-recall curve (AUC; retrieval by k-NN in the 2D space with different k's). A larger AUC is better.

7.2 Large data sets

In the second group of experiments, we tested the optimizers on four large data sets ($N \geq 58K$). It is infeasible to compute the t-SNE objective and gradients exactly due to the $O(N^2)$ cost. We thus use an established approximation technique called Barnes-Hut trees for scalable computation of cost functions and gradients [29, 36]. Conjugate gradient solvers are used for Eq. 5. It has been shown that for Neighbor Embedding the approximation loss by using Barnes-Hut trees and conjugate gradient solvers is very small [36]. The other settings are the same as for the small data sets.

The results are given in the last row of Fig. 1 and in the last four rows of Table 1. The conclusions are the same as the ones for small data sets: MM is the fastest, especially in the long run, and achieves the best mean cost function value with small standard deviations.

7.3 Over-attraction initialization

Next we show that the MM algorithm also wins when more careful initializations are used. Here we test a popular initialization strategy that amplifies the A term in the cost function by a factor $\theta > 1$ at the early iterations (see e.g. [4, 21, 30]). For many manifold em-

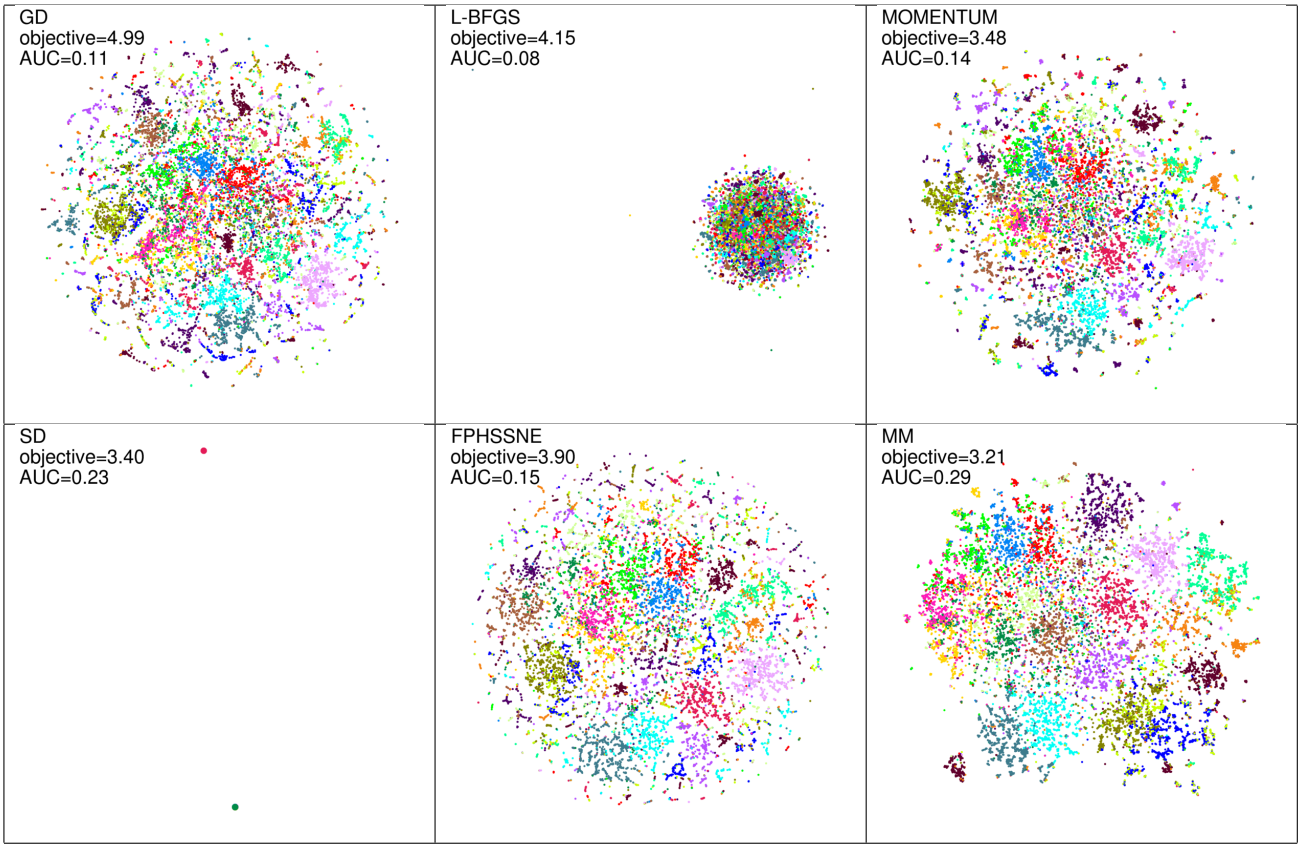

Figure 2: t-SNE visualizations of the `20NEWS` data set. Our proposed method, MM with QL-majorization, yields the best results. (The SD visualization is nearly invisible because most mapped points are crowded in the bottom, with 16 outliers drifting far away in the top, even though we use a larger marker size to increase visibility.)

bedding problems, this often leads to over-attraction at the initialization stage (also called "early exaggeration" [30]). How to choose θ and the length of initialization stage are open problems. Here we empirically used $\theta = 4$ and 30 initialization iterations, except that for MOMENTUM we used $\theta = 12$ for large datasets ($N > 10^5$) [29] and 100 initialization iterations.

We tested the compared algorithms with the over-attraction initialization on the `MNIST` data set. Figure 3 shows the evolution of the t-SNE objective over the running time. The cost decreases much faster with MM than with the other algorithms. In contrast, L-BFGS, MOMENTUM and FPHSSNE require nearly one hour to reduce the cost lower than four. SD is unstable in this case; even though it is close to MM in some runs, SD gets stuck in some plateau at early iterations. In some other runs SD is even worse, failing to reduce the cost lower than five after two hours. GD does not reduce the t-SNE cost at all with the over-attraction initialization. Table 3 shows the results of t-SNE cost function values over ten runs. It can be seen that MM achieves the best mean cost with pretty small standard deviations.

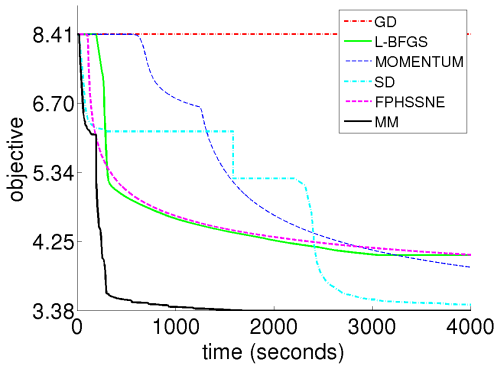

Figure 3: Evolution of the t-SNE objective (cost function value) as a function of time for the compared optimization algorithms with the over-attraction initialization for `MNIST`.

7.4 Preliminary Results beyond t-SNE

We have compared the proposed MM algorithm with other optimizers for other manifold embedding objectives (MDS-KS, s-SNE and NeRV). The preliminary results are obtained with simple random initializations

Table 2: Mean resulting manifold embedding cost function values (± standard deviation) across ten runs. Random initializations have been used. The best mean converged cost function value in each row has been boldfaced.

MDS-KS

dataname	N	GD	L-BFGS	SD	MM
COIL20	1.4K	0.43±0.08	0.84±0.78	0.15±0.02	**0.13±0.00**
20NEWS	20K	1.60±0.22	1.87±1.51	1.19±0.09	**1.11±0.03**
MNIST	70K	2.27±0.69	1.63±1.41	3.59±0.58	**1.57±0.04**

s-SNE

dataname	N	GD	L-BFGS	SD	MM
COIL20	1.4K	1.24±0.13	0.72±0.07	**0.59±0.04**	0.62±0.03
20NEWS	20K	5.78±0.23	6.47±1.00	5.31±0.04	**5.29±0.02**
MNIST	70K	6.48±1.35	6.55±1.97	**4.66±0.02**	4.68±0.03

NeRV

dataname	N	GD	L-BFGS	SD	MM
COIL20	1.4K	1.94e+03±1.71e+02	3.28e+03±3.14e+03	1.86e+03±5.38e+02	**1.33e+03±3.70e+02**
20NEWS	20K	1.45e+05±3.04e+04	1.69e+05±1.60e+04	**1.40e+05±6.84e+02**	**1.40e+05±1.24e+03**
MNIST	70K	4.91e+05±3.43e+04	5.83e+05±1.70e+05	**4.47e+05±1.68e+03**	**4.47e+05±1.69e+03**

Table 3: Resulting t-SNE objectives (mean±standard deviation) on `MNIST` with the compared optimization algorithms, using the over-attraction initialization. Boldface indicates the smallest mean objective.

Optimizer	Objective
GD	8.41±0.00
L-BFGS	3.73±0.25
MOMENTUM	3.51±0.01
SD	3.57±0.61
FPHSSNE	3.97±0.02
MM	**3.38±0.01**

on three datasets (`COIL20`, `20NEWS`, and `MNIST`). We omit FPHSSNE because currently it is not applicable to the above objectives. MOMENTUM is also omitted because it has overflow problems in most of its runs. This is probably due to a wrong learning step size, but we did not have time to solve this open problem here. For NeRV we used $\lambda = 0.9$ in $\mathcal{J}_{\text{NeRV}} = \lambda D_{\text{KL}}(P||Q) + (1-\lambda)D_{\text{KL}}(Q||P)$ and added $\epsilon = 10^{-10}$ to P to avoid log-of-zero. Caching of the Cholesky decomposition has been suggested [33], but we did not apply it on the curvature matrix because here the Cholesky factor is usually much denser than the input matrix P.

The results over ten runs of the four compared optimizers are given in Table 2. For MDS-KS, MM achieves the smallest mean cost function values across ten runs, also with much smaller standard deviations. MM and SD perform the best for s-SNE and NeRV in terms of the smallest mean and standard deviations of the resulting objective values. There are only slight differences between their converged results.

8 Conclusions and Future Work

We proposed a novel upper-bounding principle to construct Majorization-Minimization algorithms for manifold embedding problems. The proposed principle can be applied to many manifold embedding problems where a part of the cost function can be majorized by a quadratic upper bound and the rest by variable Lipschitz surrogates. In this paper and the supplemental document we provide explicit update rules for several manifold embedding methods. The resulting update rules make use of partial Hessian information, monotonically decrease cost in each iteration, and each update analytically yields the global optimum of the majorization function without needing line search.

Empirical results showed that the newly developed MM algorithms are often faster and result in better objective function values than other existing optimizers. The MM algorithms also perform robustly with different initializations.

In this work we have used simple backtracking to optimize ρ in Eq. 4. In practice we find that the average number of trials over all iterations is usually two. More comprehensive strategies with suitable heuristics could further improve the efficiency. For example, when monotonicity is not a major concern, other curvature conditions could be used to accelerate the optimization trajectory.

Acknowledgements

We thank the Academy of Finland for grants 251170, 140398, 252845, and 255725; Tekes for the Reknow project; and Yaoliang Yu for valuable discussions.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2002.

[2] D. Böhning and B. Lindsay. Monotonicity of quadratic approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40:641–663, 1988.

[3] A. Buja, D. Swayne, M. Littman, N. Dean, H. Hofmann, and L. Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, pages 444–472, 2008.

[4] M. Carreira-Perpiñán. The elastic embedding algorithm for dimensionality reduction. In *ICML*, pages 167–174, 2010.

[5] J. de Leeuw. Applications of convex analysis to multidimensional scaling. In *Recent Developments in Statistics*, pages 133–146. North Holland Publishing Company, 1977.

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[7] M. Figueiredo, J. Bioucas-Dias, and R Nowak. Majorization-Minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 16(12):2980–2991, 2007.

[8] C. Foo, C. Do, and A. Ng. A Majorization-Minimization algorithm for (multiple) hyperparameter learning. In *ICML*, pages 321–328, 2009.

[9] E. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *Graph Drawing*, volume 3383, pages 239–250, 2004.

[10] G. Hinton and S. Roweis. Stochastic neighbor embedding. In *NIPS*, pages 833–840, 2002.

[11] P. Huber. *Robust Statistics*. Wiley, New York, 1981.

[12] D. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.

[13] D. Hunter and K. Lange. Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, 9:60–77, 2000.

[14] D. Hunter and K. Lange. Computing estimates in the proportional odds model. *Annals of the Institute of Statistical Mathematics*, 54:155–168, 2002.

[15] D. Hunter and R. Li. A connection between variable selection and EM-type algorithms. Technical Report 0201, Pennsylvania State University statistics department, 2002.

[16] K. Lange and J. Fessler. Globally convergent algorithms for maximum a posteriori transmission tomography. *IEEE Transactions on Image Processing*, 4:1430–1438, 1995.

[17] K. Lange, D. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1), 2000.

[18] K. Lange and T. Wu. An MM algorithm for multicategory vertex discriminant analysis. *Journal of Computational and Graphical Statistics*, 17(3):527–544, 2008.

[19] J. Mairal. Stochastic Majorization-Minimization algorithms for large-scale optimization. In *NIPS*, pages 2283–2291, 2013.

[20] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[21] A. Noack. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480, 2007.

[22] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. We used the code at `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`.

[23] J. Ortega and W. Rheinboldt. *Iterative Solutions of Nonlinear Equations in Several Variables*, pages 253–255. New York: Academic, 1970.

[24] A. De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Transactions on Medical Imaging*, 14:132–137, 1995.

[25] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[26] C. Sabatti and K. Lange. Genomewide motif identification using a dictionary model. In *IEEE Proceedings*, volume 90, pages 1803–1810, 2002.

[27] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[28] L. van der Maaten. Fast optimization for t-SNE. In *NIPS Workshop on Challenges in Data Visualization*, 2010.

[29] L. van der Maaten. Barnes-Hut-SNE. In *ICLR*, 2013.

[30] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] L. van der Maaten, E. Postma, and J. van der Herik. Dimensionality reduction: A comparative review. Technical report, Tilburg centre for Creative Computing, Tilburg University, October 2009.

[32] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, 2010.

[33] M. Vladymyrov and M. Carreira-Perpiñán. Partial-Hessian strategies for fast learning of nonlinear embeddings. In *ICML*, pages 167–174, 2012.

[34] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70:77–90, 2006.

[35] Z. Yang, I. King, Z. Xu, and E. Oja. Heavy-tailed symmetric stochastic neighbor embedding. In *NIPS*, pages 2169–2177, 2009.

[36] Z. Yang, J. Peltonen, and S. Kaski. Scalable optimization of neighbor embedding for visualization. In *ICML*, pages 127–135, 2013.

[37] Z. Yang, J. Peltonen, and S. Kaski. Optimization equivalence of divergences improves neighbor embedding. In *ICML*, 2014.

[38] T. Zhang and G. Lerman. A novel m-estimator for robust pca. *JMLR*, 15:749–808, 2014.