NO PRESSURE! ADDRESSING PROBLEM OF LOCAL MINIMA IN MANIFOLD LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Nonlinear embedding manifold learning methods provide invaluable visual insights into a structure of high-dimensional data. However, due to a complicated nonlinear objective function, these methods can easily get stuck in local minima and their embedding quality can be poor. We propose a natural extension to several manifold learning methods aimed at identifying pressured points, i.e., points that stuck in the poor local minima and have poor embedding quality. We show that the pressure can be decreased by temporarily allowing these points to make use of an extra dimension in the embedding space. In the evaluation we show that our method is able to improve the objective function value of existing methods even after they get stuck in a poor local minimum.

1 INTRODUCTION

Given a dataset \( \mathbf{Y} \in \mathbb{R}^{D \times N} \) of \( N \) points in some high-dimensional space with dimensionality \( D \), manifold learning algorithms try to find a low-dimensional embedding \( \mathbf{X} \in \mathbb{R}^{d \times N} \) of every point from \( \mathbf{Y} \) in some space with dimensionality \( d \ll D \). These algorithms play an important role in large high-dimensional data analysis, specifically for data visualization, where \( d = 2 \) or \( d = 3 \). The quality of the methods have come a long way in recent decades, from classic linear methods (e.g., PCA, MDS), to more nonlinear spectral methods, such as Laplacian Eigenmaps (Belkin & Niyogi, 2003), LLE (Saul & Roweis, 2003) and Isomap (de Silva & Tenenbaum, 2003), finally followed by even more general nonlinear embedding (NLE) methods, which include Stochastic Neighbor Embedding (SNE, Hinton & Roweis, 2003), t-SNE (van der Maaten & Hinton, 2008), NeRV (Venna et al., 2010) and Elastic Embedding (Carreira-Perpiñán, 2010). This last group of methods is considered as state-of-the-art in manifold learning and became a go-to tool for high-dimensional data analysis in many domains (e.g., to compare the learning states in Deep Reinforcement Learning algorithms (Mnih et al., 2015) or to visualize learned vectors of some embedding model (Kiros et al., 2015)).

While the results of NLE have improved in quality, their algorithmic complexity has increased as well. PCA and most of the spectral methods are defined using a convex function with unique closed form solution. NLE methods are defined using a nonconvex objective that requires careful iterative minimization. A lot of effort has been spent on improving the convergence of NLE methods, including Spectral Direction (Vladymyrov & Carreira-Perpiñán, 2012) that uses partial-Hessian information in order to define a better search direction, or optimization using a Majorization-Minimization approach (Yang et al., 2015). However, even with these sophisticated custom algorithms, it is still often necessary to perform a few random restarts in order to achieve a decent solution. Sometimes it is not even clear whether the learned embedding represents the structure of the input data, noise, or the artifacts of an embedding algorithm (Wattenberg et al., 2016).

In this paper we focus on the analysis of the objective function of NLE methods. We are going to analyze the reasoning behind the occurrence of local minima in the objective function and ways for the algorithms to avoid them. Specifically, we discuss the conditions under which some points get caught in high-energy states of the objective function without the possibility of getting to a lower objective value. We call these points "pressured points" and show that specifically for the NLE class of algorithms there is a natural way to identify and characterize them during optimization.

Our contribution is twofold. First, we look at the objective function of the NLE methods and provide intuition and a mechanism to identify the pressured points for a given embedding. This can be used as a diagnostic tool for assessing the quality of a given embedding at the level of individual
points. Second, we propose a better optimization algorithm that is able to utilize the insights from the pressured points to find better embedding solutions. We show that our algorithm is able to get to a better objective function value even from a converged solution of an existing state-of-the-art optimizer. The proposed modification augments existing analysis of the NLE optimization and can be run on top of Spectral Direction optimization as well as $N$-body algorithms, that approximate the objective function and the gradient to run as $O(N \log N)$ (Yang et al. 2013; van der Maaten 2014) or $O(N)$ (Vladymyrov & Carreira-Perpiñán 2014).

It is important to emphasize that our analysis arises naturally from a given NLE objective function and does not depend on any other assumptions. Other papers have looked into the problem of assessing the quality of the embedding (Peltonen & Lin 2015; Lee & Verleysen 2009; Lespinats & Aupetit 2011). However, their quality criteria are defined separately from the actual learned objective function, which introduce additional assumptions and do not connect to the original objective function. We also propose a method for improving the embedding quality in addition to assessing it.

2 Nonlinear Manifold Learning Algorithms

The objective function for SNE and t-SNE was originally defined as a divergence between two probability distributions of points being in the neighborhood of each other. They both use a positive function. We also propose a method for improving the embedding quality in addition to assessing it.

It has been shown (Carreira-Perpiñán 2010; Vladymyrov & Carreira-Perpiñán 2012) that these algorithms could be defined as an interplay between two additive terms: $E(X) = E^+(X) + E^-(X)$. Attractive term $E^+$, usually convex, pulls points close to each other with a force that is larger for points located nearby in the original space. Repulsive term $E^-$, on the contrary, pushes points away from each other. Balancing and minimizing these two terms results in an embedding that preserves the overall structure of the original data.

Elastic Embedding (EE) modifies the repulsive term of the SNE objective by dropping the log, adding a weight $W^-$ to better capture non-local interactions (e.g. as $w^-_{ij} = ||y_i - y_j||^2$), and introducing a scaling hyperparameter $\lambda$ to control the interplay between two terms.

The objective functions of the described methods can be written as:

$$E_{EE}(X) = \sum_{i,j=1}^{N} w^+_{ij} \|x_i - x_j\|^2 + \lambda \sum_{i,j=1}^{N} w^-_{ij} e^{-\|x_i - x_j\|^2}, \quad (1)$$

$$E_{SNE}(X) = \sum_{i,j=1}^{N} w^+_{ij} \|x_i - x_j\|^2 + \log \sum_{i,j=1}^{N} e^{-\|x_i - x_j\|^2}, \quad (2)$$

$$E_{t-SNE}(X) = \sum_{i,j=1}^{N} w^+_{ij} \log(1 + \|x_i - x_j\|^2) + \log \sum_{i,j=1}^{N} (1 + \|x_i - x_j\|^2)^{-1}. \quad (3)$$

3 Identifying Pressured Points

Let us first consider the optimization with respect to a given point $x_0$ from $X$. For all the algorithms the attractive term $E^+$ increases as $\|x_0 - x_n\|^2$ grows and thus has a high penalty for points placed far away in the embedding space (especially if they are located nearby in the original space). The repulsive term $E^-$ is mostly localized and concentrated around individual neighbors of $x_0$. As $x_0$ navigates the landscape of $E$ it tries to get to the minimum of $E^+$ while trying to avoid the “hills” of $E^-$ created around repulsive neighbors. However, the degrees of freedom of $X$ is limited by $d$ which is typically much smaller than the intrinsic dimensionality of the data. It might happen that the point gets stuck surrounded by its non-local neighbors and is unable to find a path through.

We can illustrate this with a simple scenario involving three points $y_0, y_1, y_2$ in the original $\mathbb{R}^D$ space, where $y_0$ and $y_1$ are located nearby and $y_2$ is a bit further away. Let us try to decrease the dimensionality to $d = 1$ using EE algorithm and assume that due to poor initialization $x_0$ is located in between $x_0$ and $x_1$. In the left plot of fig. 1 we show different parts of the objective function as a function of $x_0$. The attractive term $E^+(x_0)$ creates a high pressure for $x_0$ to move towards
The function has a stationary point at $x_0$. However, the repulsion between $x_0$ and $x_2$ creates a counter pressure that pushes $x_0$ away from $x_2$. In this case, we consider $x_0$ to be a pressured point. We argue that these pressured points are the reason behind many of the local minima that arise during the minimization of NLE. Identifying those points could provide important insights into ways to improve the quality of the embeddings. By modifying and repositioning those points we can reduce the pressure and thus the overall value of the objective function.

It is important to notice the distinction between pressured points and points that just have high objective function value evaluated at that point (criteria that is used in Lespinats & Aupetit (2011) to assess the embedding quality). Not every NLE objective function can be easily evaluated for every point separately. SNE (2) and t-SNE (3) objective function contain log term that does not allow for easy decoupling. Even when such decoupling is possible (such as for EE (1)), large objective function value alone does not necessary mean that the point is stuck in a local minimum. First, the point could still be on its way to the minimum. Second, even for an embedding that represents a true global minimum there would be a variation in the objective function value between points since the affinities for every point are different.

We propose to evaluate the pressure of every point with a very simple and intuitive idea: increased pressure would create a higher energy for the point to escape that location. However, for a true local minimum of the objective function, there are no directions for that point to move. That is, given the objective function value alone does not necessary mean that the point is stuck in a local minimum. First, the point could still be on its way to the minimum. Second, even for an embedding that represents a true global minimum there would be a variation in the objective function value between points since the affinities for every point are different.

More formally, we say that the point is pressured if the objective function has a nontrivial minimum when evaluated at that point along the new dimension $Z$. We define the distance to that minimum as a pressure of that point. In what follows we are going to look at three algorithms and provide an easy way to identify and compute the pressure at a given iteration for each point.

**Elastic Embedding.** For a given point $x_k$ let us write the objective function along the new dimension $z_k$ for the EE algorithm. Notice that we consider points individually one by one, therefore all $z_{k' \neq k} = 0$. The objective function is given by:

$$\tilde{E}_{EE}(z_k) = 2z_k^2d_k^+ + 2\tilde{d}_k e^{-z_k^2} + C,$$

where $d_k^+ = \sum_{i=1}^{N} w_{ik}^+ \tilde{d}_{ik} = \lambda \sum_{i=1}^{N} w_{ik} e^{-\|x_i - x_k\|^2}$ and $C$ is a constant independent from $z_k$.

The function is symmetric wrt 0 and convex for $z_k \geq 0$. Its derivative is equal to:

$$\frac{\partial \tilde{E}_{EE}(z_k)}{\partial z_k} = 4z_k \left( d_k^+ - e^{-z_k^2} \tilde{d}_k \right).$$

The function has a stationary point at $z_k = 0$, which is a minimum when $\tilde{d}_k < d_k^+$. Otherwise, $z_k = 0$ is a maximum with non-trivial minimum given by $\hat{z}_k = \sqrt{\log(d_k^+/\tilde{d}_k)}$. The magnitude of the fraction under the log corresponds to the amount of pressure for $x_k$. The numerator $\tilde{d}_k$ depends on $X$ and represents the pressure that points in $X$ exert on $x_k$. The denominator is given by the element $k$ of the degree matrix $D^+$ and represents the attraction of the points in the original high-dimensional space. The fraction is smallest when points are ordered by $w_{ik}$ for all $i \neq k$, i.e. in

![Figure 1: Left: an illustration of the local minimum typically occurring in NLE optimization. Blue dashed lines indicate the location of 3 points in 1D. The curves show the objective function landscape wrt $x_0$. Right: by enabling an extra dimension for $x_0$, we can create a “tunnel” that avoids a local minimum in the original space, but follows a continuous minimization path in the augmented space.](image-url)
ascending order from y_k to its neighbors. As points change order and move closer to x_k (especially those far in the original space, i.e. with high w_{ik}) d_k^+ increases and eventually turns \( \bar{E}_{EE}(z_k = 0) \) from a minimum to a maximum, thus creating a pressured point.

**Stochastic Neighbor Embedding.** The objective along the new dimension is given by:

\[
\bar{E}_{SNE}(z_k) = 2z_k^2 d_k^+ + \log \left( 2(e^{-z_k^2} - 1)d_k^- + \sum_n d_n^- \right) + C, \tag{6}
\]

where, slightly abusing the notation between different methods, we define \( d_k^+ = \sum_{i=1}^N w^+_{ik} \) and \( d_k^- = \sum_{i=1}^N \exp(-\|x_i - x_k\|^2) \). The derivative is equal to

\[
\frac{\partial \bar{E}_{SNE}(z_k)}{\partial z_k} = 4z_k \left( d_k^+ - \frac{e^{-z_k^2} d_k^-}{2(e^{-z_k^2} - 1)d_k^- + \sum_n d_n^-} \right). \tag{7}
\]

Similarly to EE, the function is convex, has a stationary point at \( z_k = 0 \), which is a minimum when \( d_k^- (1 - 2d_k^+) < d_k^+ (\sum_n d_n^- - 2d_k^-) \). It also can be rewritten as \( \sum_{i=1}^N \exp(-\|x_i - x_k\|^2) < \sum_{i=1}^N w^+_{ik} \). Here, the LHS represents the pressure of the points on x_k normalized by an overall pressure for the rest of the points. If this pressure gets larger than the similar quantity in the high-dimensional space (RHS), the point becomes pressured with the minimum achieved at

\[
\dot{z}_k = \sqrt{\log \frac{d_k^+ (1 - 2d_k^+)}{d_k^- (\sum_n d_n^- - 2d_k^-)}}. \tag{8}
\]

**t-SNE.** t-SNE uses Student’s t distribution which does not decouple as nice as the Gaussian kernel in case of EE and SNE. The objective along z_k and its derivative have the following form:

\[
\bar{E}_{tSNE}(z_k) = 2\sum_{i=1}^N w_{ik} \log(1 + \|x_i - x_k\|^2 + z_k^2) + \log \left( \sum_{i,j\neq k}^N (1 + \|x_i - x_j\|^2)^{-1} + 2 \sum_{i=1}^N (1 + \|x_i - x_k\|^2 + z_k^2)^{-1} \right) + C. \tag{9}
\]

\[
\frac{\partial \bar{E}_{tSNE}(z_k)}{\partial z_k} = 4z_k \left( \sum_{i=1}^N \frac{w^+_{ik}}{1 + \|x_i - x_k\|^2 + z_k^2} - \frac{\sum_{i=1}^N (1 + \|x_i - x_k\|^2 + z_k^2)^{-2}}{\sum_{i,j\neq k}^N (1 + \|x_i - x_j\|^2)^{-1} + 2 \sum_{i=1}^N (1 + \|x_i - x_k\|^2 + z_k^2)^{-1}} \right). \tag{10}
\]

It is not easy to find the minimum in a closed form, however practically it can be done with just few iterations of the Newton’s method initialized at some positive value close to 0. In addition, we can quickly test whether the point is pressured or not from the sign of the second derivative at \( z_k = 0 \):

\[
\frac{\partial^2 \bar{E}_{tSNE}(0)}{\partial^2 z_k} = \sum_{i=1}^N \frac{w^+_{ik}}{1 + \|x_i - x_k\|^2} - \frac{\sum_{i=1}^N (1 + \|x_i - x_k\|^2)^{-2}}{\sum_{i,j=1}^N (1 + \|x_i - x_j\|^2)^{-1}}. \tag{11}
\]

## 4 Pressured Points for Embedding Quality Analysis

The analysis above can be directly applied to the existing algorithms as is, resulting in a qualitative statistic of the amount of pressure each point is experiencing during optimization. A nice additional property is that computing pressure points for each algorithm can be done in constant time by reusing parts of the gradient. A practitioner can run the analysis for every iteration of the algorithm essentially for free to see how many points are pressured and whether the embedding can be trusted.

In fig[2] we show a couple of the examples of embeddings with pressured points computed. The embedding of the swissroll on the left had a poor initialization that SNE was not able to recover from. Pressured points are concentrated around the twist in the embedding and in the corners, precisely where the difference with the ground truth occurs. On the right, we can see the embedding of the subset of COIL-20 dataset midway through optimization with EE. The dataset consists of photos of 10 different objects as they are rotated on a platform with new photo taken every 5 degrees (72 data points per object). Good embedding should separate objects one from another and also reflect the rotational sequence for each object. However, the embeddings of some objects overlap with each others, which results in high pressure.
Figure 2: Some examples of pressured points for different datasets. Larger marker size corresponds to the higher pressure value. Color corresponds to the ground truth. Left: SNE embedding of the swissroll dataset with poor initialization that results in a twist in the middle of the roll. Right: 10 objects from COIL-20 dataset after 100 iteration of EE. Color encodes different objects and each point corresponds to 2D embedding of one image of a given object.

In fig.3 we show an embedding of the subset from MNIST after 200 iteration of t-SNE. We highlight some of the digits that ended up in clusters that are different from their ground truth. We put them in a red frame if a digit has a high pressure and in a green frame if their pressure is 0. For the most part the digits in red squares don’t belong to clusters where they are currently located, while digits in green squares look very similar to the digits around them.

5 IMPROVING CONVERGENCE BY PRESSURED POINTS OPTIMIZATION

The analysis above can be independently used for assessing the embedding quality, but it can be used for improvements in optimization as well. Imagine for some iteration $X$ we have a set of points $P$ that are pressured according to the definition above. Effectively it means that given a new dimension $d+1$ the pressured points would utilize it in order to escape some local minima. Let us create this new dimension $Z$ with $z_k \neq 0$ for all $k \in P$. Non-pressured points still exist in the same space $d$ and can only move along those $d$ dimensions. For example, augmented objective function for EE would look like this:

$$\tilde{E}(X, Z) = E(x_{j \notin P}) + E\left( \left( \frac{x_i}{z_i} \right)_{i \in P} \right) +$$

$$+ 2 \left( \sum_{i \in P} \sum_{j \notin P} w_{ij}^+ \|x_i - x_j\|^2 + \sum_{i \in P} z_i^2 \sum_{j \in P} w_{ij}^- + \lambda \sum_{i \in P} e^{-z_i^2} \sum_{j \notin P} w_{ij}^- e^{-\|x_i - x_j\|^2} \right).$$

(12)

The objective function splits into three parts: interaction between non-pressured points $X$ in $d$-dimensional space, interaction between pressured points for $X$ and $Z$ in $d+1$ dimensional space and interaction between pressure and non-pressure points. The first two parts essentially represent the minimization of pressured and non-pressured points independently in $d$ and $d+1$ dimensional space.
The last part is quite interesting. The first two of out of tree terms come from the attractive part and the third is from the repulsive part. The first term represents the interaction between pressured and non-pressured points X in d space. Second term essentially pulls each $z_i$ to 0 with the weight proportional to the attraction between point $i$ and all the non-pressured points. Notice that its form is identical to the $l_2$ norm applied to the extended dimension Z with the weight given by the attraction between point $i$ and all the non-pressured points. Finally, the last term captures the interactions between Z coordinate of pressured points and X space for non-pressured points. On one hand, it pushes Z away from 0 as pressured and non-pressured points move closer to each other in d space. On the other hand, it re-weights the repulsion between pressured and non-pressured points proportional to $\exp(-z_i^2)$, which means that the repulsion gets smaller for larger values of $z_i$. In fact, since $\exp(-z^2) < 1$ for all $z > 0$, the repulsion between pressured and non-pressured points would always be weaker than the repulsion of non-pressured points between each other.

Since our final objective is not to find an exact minimum of (12), but rather a better embedding of X, we are going to add few additional steps to each iteration of the optimization. First, the validity of the set of pressured points. After each iteration of minimizing (12) we are going to remove points from $P$ that are not pressured anymore and check X for new pressured points and add them to $P$. Second, we want pressured points to explore the new dimension only to the extent that it could eventually help lowering the value of the original objective function. We want to restrict the use of the new dimension so that it is not treated in the same way as the other ones. The most natural would be to gradually increase the cost for the points to use the new dimension. It could be achieved by adding $l_2$ penalty to Z dimension as $\mu \sum_{i \in P} z_i^2$. This is an organic extension since it has the same form as the second term in (12). For $\mu = 0$ the penalty is given naturally by the weight between pressured and non-pressured points and for larger $\mu$ the penalty eventually becomes too large to utilize the new dimension at all. This property provides a distinct advantage for our algorithm comparing to the standard use of $l_2$ regularization. In a typical scenario a practitioner has little control over the effect of different $\mu$ values and has to resort to trial and error of different values to find a sweet spot where the regularization has a desirable effect. In our case, the regularizer already exists in the objective and its weight sets a natural scale of $\mu$ values to try. Another advantage over traditional regularizer is that there is no upper bound of $\mu$ beyond which the algorithm stops working. For large $\mu$ the points along Z just collapse to 0 and the algorithm falls back to the original one.

Practically, we propose to use a sequence of $\mu$ values starting at $\mu = 0$, for which the extra dimension can be fully utilized. We increase $\mu$ proportionally to the magnitude of $d_k^T$, $k = 1 \ldots N$. In the experiments below, we set step = $1/N \sum_k d_k^T$, although a more aggressive schedule of step = $\max(d_k^T)$ or more conservative step = $\min(d_k^T)$ can be used. We increase $\mu$ up until $z_k = 0$ for all the points. Typically, it occurs after 4–5 steps over $\mu$.

The resulting method is described in Algorithm 1. The algorithm can be run on top of the existing optimization methods for NLE: Spectral Direction and N-body methods.

For Spectral Direction, the Hessian is approximated using the second derivative of $E^+$ only. The search direction has the form $P = (4L^+ + \epsilon)^{-1}G$, where G is the gradient, $L^+$ is the graph Laplacian defined on positive affinities $W^+$ and $\epsilon$ is a small constant that makes the inverse possible. The modified objective that we propose has one more quadratic term $\mu ZZ^T$ and thus the Hessian for the pressured points along Z dimension is regularized by a positive value $2\mu$. This is good for two reasons: it improves the direction of the Spectral Direction by adding new bits of Hessian, and it makes the Hessian approximation pd, thus avoiding the need to add any constant to it.

Large-scale N-Body approximations using Barnes-Hut (Yang et al. 2013; van der Maaten 2014) or Fast Multipole Methods (FMM, Vladymyrov & Carreira-Perpinan, 2014) decrease the cost of objective function and the gradient from $O(N^2)$ to $O(N \log N)$ or $O(N)$ by approximating the interaction between distant points. Pressured points computation uses the same quantities as the gradient, so whichever approximation is applied to the gradient could also be applied to compute pressured points. The only difference comes from slightly increased computational cost of minimizing (12) which includes a term that involves computation between pressured points in $d + 1$ dimension. However, this affects only the computation between the pressure points and, practically, we did not observe much of a difference in the runtime.
Finally, in fig. 5 we show the minimization of MNIST using FMM approximation with \( p = 5 \) accuracy (i.e. truncating the Hermite functions to 5 terms). PP optimization improved the convergence of the objective function, but we are going to call it simply PP to avoid extra clutter.

\[ \text{Algorithm 1: Pressure Points Optimization using Extra Dimension} \]

**Input**: Initial \( X \), sequence of regularization steps \( \mu \).

Compute a set of pressured points \( P \) from \( X \) and initialize \( Z \) according to their pressure value.

\[ \text{foreach } \mu_i \in \mu \] 

\[ \text{repeat} \]

update \( X, Z \) using Spectral Direction over \( \min (\tilde{E}(X, Z) + \mu_i ZZ^T) \).

Update \( P \) using pressured points computed from new \( X \):

1. Add new points to \( P \) according to their pressure value.
2. Remove points that are not pressured anymore.

\[ \text{until convergence; } \]

**Output**: final \( X \)

6 Experiments

Here we are going to compare the original unmodified algorithm, which we call simply spectral direction (SD) to the Pressure Point (PP) optimization framework that we have defined above.\(^{1}\)

We applied the algorithm to EE and SNE methods. While the proposed methodology could also be applied to \( t \)-SNE, in practice we were not able to find it useful for this method. \( t \)-SNE is defined on Student’s \( t \) kernel that has much longer tails than the Gaussian kernel used in EE and SNE. Because of that, the repulsion between points is much stronger and points are spread far away from each other. The extra-space given by new dimension is not utilized well and the objective function decrease is similar with and without the PP modification.

We first run the algorithm on 10 objects from COIL-20 dataset. We run both SNE and EE 10 different times with the original unmodified algorithm till the objective function does not change for more than \( 10^{-5} \) per iteration. We then run PP optimization with two different initializations. First, we use exact same initialization as the original algorithm to check which algorithm gets a lower score from the same starting point. Second, we initialize PP from the convergence value of SD to see if we could get out of the local minima and improve the results. Over 10 runs for EE, SD got to an average objective function value of \( 3.84 \pm 0.18 \), whereas PP with random initialization got to \( 3.6 \pm 0.14 \). Initializing from the convergence of SD, 10 out of 10 times PP was able to find better local minima with the average objective function value of \( 3.61 \pm 0.19 \). We got similar results for SNE: average objective function value for SD is \( 11.07 \pm 0.03 \), which PP improved to \( 11.03 \pm 0.02 \) for random initialization and to \( 11.05 \pm 0.03 \) for initialization from local minima of SD. In fig. 6, we show the results for one of the runs for EE and SNE as well as the final embedding after SD and PP optimization. Notice that for initial small \( \mu \) values the algorithm extensively uses and explores the extra dimension, which one can see from increase in the original objective function values as well as from the large fraction of the pressured points. However, for larger \( \mu \) the number of pressured points drops sharply, eventually going to 0. Once \( \mu \) gets large enough so that extra dimension is not used, optimization for every new \( \mu \) goes very fast, since essentially nothing is changing. On the right side we show the local minimum of SD and the better minimum that was found by PP. Notice that some of the individual digits (e.g. yellow, light blue and top green) improved their embedding. The number of pressured points and the sum of all the pressured values have also decreased.

As another comparison point, we evaluate how much headroom we can get on top improvements demonstrated by PP algorithm. For that, we run EE with homotopy method (Carreira-Perpiñán, 2010) where we performed a series of optimizations from a very small \( \lambda \), where the objective function has a single global minimum, to final \( \lambda = 200 \), each time initializing from the previous solution. We got the final value of the objective function around \( E = 3.28 \) (dashed red line on the EE objective function plot). While we could not get to a same value with PP, we got very close with \( E = 3.3 \) (comparing to \( E = 3.68 \) for the best SD optimization).

Finally, in fig. 5 we show the minimization of MNIST using FMM approximation with \( p = 5 \) accuracy (i.e. truncating the Hermite functions to 5 terms). PP optimization improved the convergence

\(^{1}\)It would be more fair to call our method SD+PP, since we also apply spectral direction to minimize the extended objective function, but we are going to call it simply PP to avoid extra clutter.
Figure 4: EE (left) and SNE (center) optimization for 10 objects from COIL-20 dataset using SD (black) and PP optimization using random initialization (green) and initialization from the local minima of the SD (blue). Top plots show the change in the original unmodified objective function, while the bottom shows the fraction of the points that are pressured for any given iteration. Markers ‘o’ indicate change of μ value. Dashed red line indicate the absolute best value of the objective function that we were able to get with very careful homotopy optimization. Right: SNE embedding for SD (top) and PP (bottom) when initialized from the local minima.

Figure 5: Optimization of MNIST dataset using SD and PP with random initialization (green curve) and from the minimum of SD (blue curve). Center and right plot show the difference in the convergence of SD and PP when initialized from the minimum of SD. The size of the markers correspond to the amount of pressure for each points.

both in the case of random initialization and in case of initialization from the local minimum of SD. This demonstrates that the benefits of PP algorithm can be increased by also applying SD to improve the optimization direction and FMM to speed up the objective function and gradient computation.

7 CONCLUSIONS

We proposed a novel framework for assessing the quality of several manifold learning methods using intuitive, natural and computationally cheap way to measure the pressure that each point experiencing from its neighbors. We then outlined the method to make use of that extra dimension in order to find a better solution for the pressured points. The algorithm works well for EE and SNE methods when initialized at random or from the local minima of some other methods. An interesting future direction would be to extend this analysis to measure intrinsic dimensionality of the manifold. In other words, answer the question: how many dimensions do we need to remove all the pressured points?


REFERENCES


