Multidimensional Scaling in the Poincaré Disk

Andrej Cvetkovski
Boston University
acvetk@gmail.com

Mark Crovella
Boston University
crovella@cs.bu.edu

Abstract—**Multidimensional scaling (MDS) is a class of projective algorithms traditionally used to produce two- or three-dimensional visualizations of datasets consisting of multidimensional objects or interobject distances. Recently, metric MDS has been applied to the problems of graph embedding for the purpose of approximate encoding of edge or path costs using node coordinates in metric space. Several authors have also pointed out that for data with an inherent hierarchical structure, hyperbolic target space may be a more suitable choice for accurate embedding than Euclidean space. In this paper we present the theory and the implementation details of MDS-PD, a metric MDS algorithm designed specifically for the Poincaré disk model of the hyperbolic plane. Our construction is based on an approximate hyperbolic line search and exemplifies some of the particulars that need to be addressed when applying iterative optimization methods in a hyperbolic space model. MDS-PD can be used both as a visualization tool and as an embedding algorithm. We provide several examples to illustrate the utility of MDS-PD.**

Index Terms—**dimensionality reduction, hyperbolic embedding, hyperbolic MDS, network graph, steepest descent, visualization**

I. INTRODUCTION

Metric multidimensional scaling (MDS) (Carroll and Arabie 1980; De Leeuw and Heiser 1982; Cox and Cox 2000; Borg and Groenen 2005) is a class of algorithms that take as input some or all of the inter-object distances (*pair dissimilarities*) for n objects and produce as output a *point configuration* of n points specified by their coordinates in a chosen d-dimensional *target space*. The goal is to return the point configuration whose inter-point distances in the d-dimensional space match as closely as possible the original input distances. Usually, this goal is pursued by minimizing a scalar badness-of-fit *objective function* defined for an arbitrary n-point configuration in the target space; ideally, the output of an MDS algorithm should be the configuration that achieves the global minimum of the objective function.

If the target space dimension is 2 or 3, the output configuration can be graphically represented, which makes MDS a visualization tool seeking to preserve the input distances as faithfully as possible, thus clustering the objects in the target space by similarity. More generally, for a given dimension d, metric multidimensional scaling can be used to *embed* an input set of dissimilarities of the original objects into a d-dimensional metric space.

In order to apply MDS, several design decisions must be made. One first needs to choose a target metric space of appropriate dimension d and a corresponding distance function. An objective function should be chosen so that it provides a suitable measure of inaccuracy for a given embedding application. If the objective function is nonlinear but satisfies some mild general conditions (smoothness), a numerical optimization method can be chosen (e.g. Nocedal and Wright 1999) for the implementation of the embedding algorithm.

The Euclidean plane is the most common choice of target space for visualization applications due to its simplicity and intuitiveness. Spherical surface can be used, for example, to avoid the edge effect of a planar representation (Cox and Cox 1991). In general, MDS on curved subspaces of Euclidean space can be viewed as MDS in a higher dimensional Euclidean space constrained to a particular surface (Bentler and Weeks 1978; Bloxom 1978).

The use of metric MDS in the hyperbolic plane was proposed in the context of interactive visualization by Walter (2004), inspired by the focus and context hyperbolic tree viewer of Lamping and Rao (1994). The use of MDS in (Walter 2004) inherently generalized the applicability of the method from tree structures to continuous-valued multidimensional data or pair distances. It was noted that the curious property of exponential growth of the "available space" in the hyperbolic models as

one moves towards infinity, makes planar hyperbolic embedding a suitable choice for both hierarchical and high-dimensional data. The adequacy of the hyperbolic spaces for embedding of various data was also studied and confirmed in the contexts of network embedding for path cost estimation (Shavitt and Tankel 2008) and routing (Kleinberg 2007; Krioukov et al. 2008; Cvetkovski and Crovella 2009).

A least squares formulation of MDS, to be used in conjunction with an iterative numerical method for unconstrained optimization was proposed by Sammon (1969). The objective function therein (the Sammon stress criterion) was defined as a normalized sum of the squared differences between the original dissimilarities and the embedded distances of the final point configuration. To minimize this function, Sammon proposed a descent method with step components calculated using the first two component derivatives of the objective function. Walter (2004) adopted Sammon's badness-of-fit measure for hyperbolic MDS but observed that applying Sammon's iterative procedure in the Poincaré disk (PD) using exact derivatives is difficult due to the complicated symbolic expressions of the second derivative of the hyperbolic distance function in this model. Subsequently, the Levenberg-Marquardt least squares method was applied in (Walter 2004), using only first-order derivatives for the optimization, but the details of applying this iterative method in the Poincaré disk were not elaborated.

In this paper we present MDS-PD, a metric multidimensional scaling algorithm using the Poincaré disc (PD) model. MDS-PD is based on a steepest decent method with line search. We show the details of the steepest descent along hyperbolic lines in the PD and present a suitable approximate hyperbolic line search procedure. MDS-PD is applicable in its own right; additionally, its construction also illustrates some of the specifics that need to be considered when transferring more sophisticated iterative optimization methods to the PD or to other hyperbolic models. Based on this development, we show the particulars of a numerical implementation of MDS-PD and use this algorithm to carry out numerical experiments on several datasets using least squares criterion functions. Our simulation results indicate that the performance of a steepest descent method for minimizing a least squares objective on large configurations in the PD is notably dependent on the line search method used, and that binary hyperbolic line search provides markedly better convergence and cost properties for MDS-PD compared to more sophisticated or precise methods.

The rest of this paper is organized as follows. Section II consolidates the notation and concepts from hyperbolic geometry that will be used throughout, and proceeds to develop two of the building blocks of MDS-PD – steepest descent in the PD and a corresponding hyperbolic line search. Section III considers particular objective functions and gradients and further discusses properties and applicability of multidimensional scaling in the PD. Section IV presents illustrative examples of MDS-PD operation in the context of synthetic as well as real-world input data. Related work is discussed in Section V and concluding remarks are given in Section VI.

II. A DESCENT METHOD FOR THE POINCARÉ DISK

We start this section by introducing our notational conventions and establishing some properties of the Poincaré disk that will be useful in the subsequent development. This will allow us to formally define a Poincaré-disk specific descent method and a binary hyperbolic line search algorithm, that together make a simple, yet efficient iterative minimization method for this model of the hyperbolic plane.

A. Preliminaries

The *Poincaré Disk model* of the hyperbolic plane is convenient for our considerations since it has circular symmetry and a closed form of the inter-point distance formula exists (Anderson 2007). We will be using complex rectangular coordinates to represent the points of the hyperbolic plane, making the PD model a subset of the complex plane $\mathbb{C}$:

$$\mathbb{D} = \{z \in \mathbb{C} \mid |z| < 1\}. \tag{1}$$

The *hyperbolic distance* between two points z_j and z_k in $\mathbb{D}$ is given by

$$d_{\mathbb{D}}(z_j, z_k) = 2\mathrm{atanh}\frac{|z_j - z_k|}{|1 - z_j\overline{z_k}|}, \tag{2}$$

where $\bar{z}$ denotes the complex conjugate.

Möbius transformations are a class of transformations of the complex plane that preserve generalized circles. The special Möbius transformations that take $\mathbb{D}$ to $\mathbb{D}$ and preserve the hyperbolic distance have the form

$$T(z) = \frac{az+b}{\overline{b}z+\overline{a}}, \ \ a,b \in \mathbb{C}, \ \ |a|^2 - |b|^2 \neq 0. \tag{3}$$

Given a point $z_0 \in \mathbb{D}$ and a direction $\gamma \in \mathbb{C}$ with $|\gamma| = 1$, we can travel a hyperbolic distance $s \geq 0$ along

a hyperbolic line starting from z_0 in the direction γ, arriving at the point z_0'.

Lemma 1. *For $z_0 \in \mathbb{D}$, $\gamma \in \mathbb{C}$ with $|\gamma| = 1$, and $s \geq 0$, the point*

$$z_0' = \frac{\gamma \tanh \frac{s}{2} + z_0}{\overline{z_0} \gamma \tanh \frac{s}{2} + 1}$$

(i) belongs to the hyperbolic ray passing through z_0 and having direction γ at z_0, and
(ii) $d_{\mathbb{D}}(z_0, z_0') = s$.

Proof: Given a point $z_0 \in \mathbb{D}$ and a direction $\gamma \in \mathbb{C}$ with $|\gamma| = 1$, the hyperbolic ray in $\mathbb{D}$ passing through z_0 and having direction γ at z_0 can be parametrized by $r \in [0, 1)$ as

$$f(r) = \frac{r\gamma + z_0}{r\gamma \overline{z_0} + 1}. \tag{4}$$

Noting that (4), seen as a function of $z = r\gamma$:

$$T(z) = \frac{z + z_0}{z\overline{z_0} + 1}$$

is a Möbius transformation taking $\mathbb{D}$ to $\mathbb{D}$ and preserving hyperbolic distances, we see that

$$d_{\mathbb{D}}(f(r), z_0) = d_{\mathbb{D}}(0, r) = \ln \frac{1+r}{1-r}$$

whence it follows that moving z_0 along a hyperbolic line in the direction γ by a hyperbolic distance $s = \ln((1+r)/(1-r))$ we arrive at the point $z_0' = f\left(\tanh \frac{s}{2}\right)$. ∎

Next, we introduce some of the notation that will be used throughout.

- Let the *point configuration* at iteration $t = 1, 2, \ldots T$ consist of n points in the Poincaré disk $\mathbb{D}$

$$z_j(t), \quad j = 1 \ldots n$$

  represented by their rectangular coordinates:

$$z_j(t) = y_{j,1}(t) + iy_{j,2}(t), i = \sqrt{-1}, y_{j,1}, y_{j,2} \in \mathbb{R}$$

  with $|z_j(t)| < 1$.
- We also use vector notation to refer to the point configuration

$$\mathbf{z}(t) = \begin{bmatrix} z_1(t) & z_2(t) & \ldots & z_n(t) \end{bmatrix}^T = \mathbf{y}_1 + i\mathbf{y}_2 =$$
$$= \begin{bmatrix} y_{1,1}(t) & y_{2,1}(t) & \ldots & y_{n,1}(t) \end{bmatrix}^T +$$
$$i \begin{bmatrix} y_{1,2}(t) & y_{2,2}(t) & \ldots & y_{n,2}(t) \end{bmatrix}^T,$$

  where $[\cdot]^T$ in this work indicates the real matrix transpose (to be distinguished from the complex conjugate transpose.)
- The *distance matrix* for a given point configuration $\mathbf{z}$ is the real valued symmetric matrix $\mathbf{D}(\mathbf{z}) = [d_{jk}]_{n \times n}$ whose entry d_{jk} is the hyperbolic distance between points z_j and z_k in the configuration $\mathbf{z}$:

$$d_{jk} = d_{\mathbb{D}}(z_j, z_k).$$

- The *dissimilarity matrix* $\mathbf{\Delta} = [\delta_{jk}]_{n \times n}$ is a symmetric, real-valued matrix containing the desired inter-point distances of the final output configuration (the *dissimilarities*). The diagonal elements are $\delta_{jj} = 0$ and all other entries are positive real numbers: $\delta_{jk} = \delta_{kj} > 0$ for $j \neq k$.
- The *indicator matrix* $\mathbf{I} = [I_{jk}]_{n \times n}$ is a symmetric 0-1 matrix, used to allow for missing dissimilarity values. The entries of $\mathbf{I}$ corresponding to missing values in $\mathbf{\Delta}$ are set to 0. All other entries are set to 1.
- The *weight matrix* $\mathbf{W} = [w_{jk}]_{n \times n}$ is a symmetric, real-valued matrix introduced to enable weighting of the error terms for individual pairs of points in the objective function sum. For convenience, w_{jk} corresponding to missing dissimilarities are set to some finite value, e.g. 1.
- The *objective function* to be minimized is the *embedding error function* $E_t = E_t(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$ that, given the sets of dissimilarities and weights, associates to a configuration $\mathbf{z}$ an embedding error E_t. An example of an error function is the sum of relative squared differences

$$E_t(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I}) = \sum_{j=1}^{n} \sum_{k=j+1}^{n} w_{jk} I_{jk} \left(\frac{d_{jk}(t) - \delta_{jk}}{\delta_{jk}} \right)^2. \tag{5}$$

  The objective function can optionally be normalized per pair by dividing with the number of summands $(n^2 - n)/2$.

B. Descent in the Poincaré disk

Given a configuration of points $\mathbf{z}$, matrices $\mathbf{\Delta}$, $\mathbf{W}$, and $\mathbf{I}$, the distance function $d_{\mathbb{D}}(z_j, z_k)$, and an objective function $E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$, define

$$\mathbf{g} = \nabla E \overset{\text{def}}{=} \begin{bmatrix} \frac{\partial E}{\partial y_{1,1}} + i\frac{\partial E}{\partial y_{1,2}} \\ \frac{\partial E}{\partial y_{2,1}} + i\frac{\partial E}{\partial y_{2,2}} \\ \vdots \\ \frac{\partial E}{\partial y_{n,1}} + i\frac{\partial E}{\partial y_{n,2}} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}. \tag{6}$$

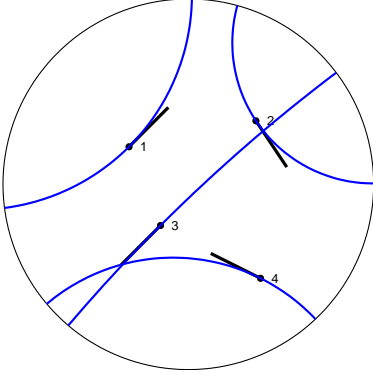According to Lemma 1, moving the points $z_1, \ldots, z_n$ of

Figure 1. An example of moving a 4-point configuration in a given (descent) direction along distance realizing paths of the Poincaré disk

the configuration $\mathbf{z}$ along distance realizing paths in the PD defined respectively by the directions $-g_1, \ldots, -g_n$ at $\mathbf{z}$ (Fig. 1) will result in configuration $\mathbf{z}'$ with points

$$z'_j = \frac{-rg_j + z_j}{-rg_j \overline{z_j} + 1} \qquad (7)$$

where $r \geq 0$ is the *step-size parameter* which determines the hyperbolic distances s_j traveled by z_j:

$$s_j = \ln \frac{1 + r|g_j|}{1 - r|g_j|}. \qquad (8)$$

The PD model (1) implies the constraints $|z_j| < 1$ for the point coordinates. Still, the optimization on the PD can be viewed as unconstrained by observing that the constraints $|z'_j| < 1$ will not be violated while moving a configuration $\mathbf{z}$ in $\mathbb{D}$ if the distances s_j traveled by each point are always kept finite, i.e.

$$s_M = \max_j s_j < \infty. \qquad (9)$$

Since (9), according to (8), corresponds to $r \max_j |g_j| < 1$, we have the constraint on r

$$r < \frac{1}{\|\mathbf{g}\|_\infty}.$$

When implementing iterative descent minimization methods with line search in the Poincaré disk, it is important to specify a hyperbolic distance window s_M along the descent lines where the next configuration will be sought. In this case the corresponding value of the parameter r is

$$r_M = \frac{1}{\|\mathbf{g}\|_\infty} \cdot \tanh \frac{s_M}{2} < \frac{1}{\|\mathbf{g}\|_\infty}. \qquad (10)$$

Since the Poincaré disk model is conformal, following the direction $-\mathbf{g}$ (the opposite of (6)) corresponds to the *steepest descent* optimization method. Moving the point

configuration along hyperbolic lines (distance realizing paths), on the other hand, ensures that the steepest descent direction is exhausted most efficiently given the current information about the objective function.

C. A Steepest Descent Algorithm for the PD

Figure 2 shows a framework for MDS-PD.

Algorithm MDS-PD

Input data:
  an initial configuration $\mathbf{z}(1)$
  the dissimilarities $\mathbf{\Delta}$, weights $\mathbf{W}$, indicators $\mathbf{I}$
Input parameters:
  an objective function $E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$
  the stopping tolerances ε_E, $\varepsilon_{\Delta E}$, $\varepsilon_{\mathbf{g}}$, ε_r, T_M
Output:
  a final point configuration $\mathbf{z}(T)$
  a final embedding error E_T

Initialize:
  $t \leftarrow 1$; $s_M \leftarrow 10$; $E_{-1} \leftarrow \infty$; $\mathbf{z} \leftarrow \mathbf{z}(1)$; .... {2.1}
Loop:
  $E \leftarrow E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$; $\mathbf{g} \leftarrow \nabla E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$; ..... {2.2}
  $r_M \leftarrow \frac{1}{\|\mathbf{g}\|_\infty} \cdot \tanh \frac{s_M}{2}$; .... {2.3}
  Break if
$$\left.\begin{array}{ll} & E < \varepsilon_E \\ \mathbf{or} & E_{-1} - E < \varepsilon_{\Delta E} \\ \mathbf{or} & \|\mathbf{g}\|_\infty < \varepsilon_{\mathbf{g}} \\ \mathbf{or} & r_M < \varepsilon_r \\ \mathbf{or} & t > T_M; \end{array}\right\} \ldots\ldots\ldots\ldots\ \{2.4\}$$
  $E_{-1} \leftarrow E$;
  $r \leftarrow \text{HypLineSearch}(E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I}), -\mathbf{g}, 0, r_M)$; . {2.5}
  $\forall j \in \{1..n\}, \ z_j \leftarrow \frac{-rg_j + z_j}{-rg_j \overline{z_j} + 1}$; .... {2.6}
  $t \leftarrow t + 1$;
Return $\mathbf{z}(T) \leftarrow \mathbf{z}$ and $E_T \leftarrow E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$.

Figure 2. MDS-PD

The input data of MDS-PD consists of the initial configuration $\mathbf{z}(1)$, and the input metric: the dissimilarities $\mathbf{\Delta}$ with the associated weights $\mathbf{W}$ and the indicators of missing dissimilarities $\mathbf{I}$. The input parameters are the objective error function $E(\mathbf{z}, \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$ and the stopping tolerances ε_E, $\varepsilon_{\Delta E}$, $\varepsilon_{\mathbf{g}}$, ε_r, and T_M. The output of MDS-PD consists of the final point configuration $\mathbf{z}(T)$ and its associated embedding error $E_T = E(\mathbf{z}(T), \mathbf{\Delta}, \mathbf{W}, \mathbf{I})$.

The initialization {2.1} sets the maximum hyperbolic distance s_M that can be traveled by any point of the configuration, and the previous value of the embedding error E_{-1}.

Each iteration starts by determining the gradient of the error in the current configuration {2.2} and the corresponding window r_M {2.3} for the parameter r (Eq. (10)). A hyperbolic line search (described in Sec. II-D) is performed {2.5} in the direction of the steepest descent $-\mathbf{g}$ of the embedding error and the resulting step-size parameter r is used in {2.6} to arrive at the next configuration as in (7).

Several stopping criteria are used (line {2.4}) to terminate the search. Ideally, the algorithm exits when the embedding error is close to 0 ($E < \varepsilon_E$). Termination also occurs in the cases when the error decreases too slowly ($E_{-1} - E < \varepsilon_{\Delta E}$), or when the gradient or the step-ping parameter become too small ($\|\mathbf{g}\|_\infty < \varepsilon_\mathbf{g}$, $r_M < \varepsilon_r$). Finally, T_M, the maximum allowed number of iterations, is used as a guard against infinite looping.

The line search subprogram used in {2.5} is described next.

D. Approximate Hyperbolic Line Search

An *exact line search* could be used in line {2.5} (Fig. 2) to determine a value for the step size r such that the corresponding new configuration {2.6} achieves a local minimum of the embedding error along the search path with tight tolerance:

$$r \approx \mathrm{argmin}_{r \in [0, r_M]} q(r), \qquad (11)$$

where $q(r)$ is the embedding error as a function of r.

However, increasing the precision of this computation is not essential to the convergence performance since the steepest descent search direction is only locally optimal. Furthermore, exact line search can fail to converge to a local minimum even for a second degree polynomial due to finite machine precision (Frandsen et al. 2004). It is now accepted in the numerical optimization literature that *approximate line search* provides convergence rates comparable to the exact line search while significantly reducing the computational cost per line search. In fact, the step calculation used in Sammon (1969) is a "zero-iteration" approximate line search, where the step size is simply guessed based on the first two derivatives of the error. Conceivably, the simplest inexact step calculation would guess the step size based only on the directional gradient at the current configuration.

Approximate line search procedures aim to reduce the computational cost of determining the step parameter by posing weaker conditions on the found solution: Rather than searching for a local or global minimizer of $q(r)$
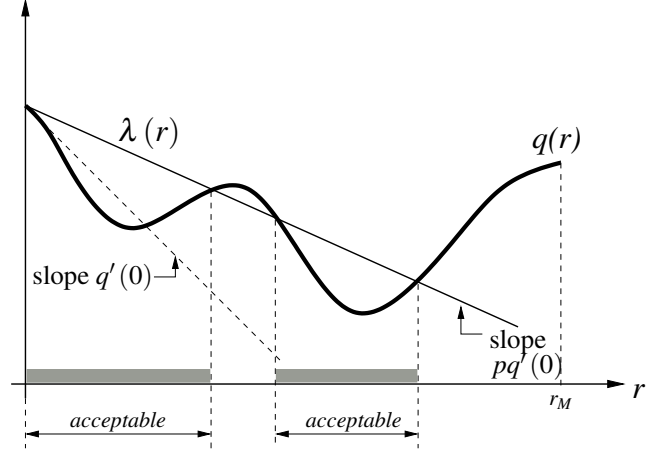

Figure 3. Acceptable step lengths for inexact line search obtained from the sufficient decrease condition.

on $(0, r_M]$, a value is returned by the line search function as satisfactory if it provides sufficient decrease of the objective function and sufficient progress toward the solution configuration. A popular approach to defining sufficient decrease is to define the "roof" function

$$\lambda(r) = q(0) + p \cdot q'(0) \cdot r, \quad 0 < p < 1 \qquad (12)$$

which is a line passing through $(0, q(0))$ and having a slope which is a fraction of the slope of $q(r)$ at $r = 0$. With this function, we define that sufficient decrease is provided by all values of r such that

$$q(r) < \lambda(r), \ r \in (0, r_M] \qquad (13)$$

Fig. 3 shows an example of acceptable step length segments obtained from the sufficient decrease condition (13).

To ensure sufficient progress, we adopt a binary search algorithm motivated by the simple backtracking approach (e.g. Nocedal and Wright (1999)). The details are given in Fig. 4.

We start the line search with an initial guess r_0 for the step size parameter, and in the expansion phase {4.1} we double it until it violates the window r_M or the sufficient decrease condition. In the reduction phase {4.2}, we halve r until it finally satisfies both the window requirement $r < r_M$ and the decrease criterion $q(r) < \lambda(r)$.

We observe that, when started at a point with nonzero gradient, the line search will always return a nonzero value for r. Since the returned acceptable step r is such that the step $2 \cdot r$ is not acceptable, there will be a maximum acceptable point r_m from the same acceptable

Procedure HypLineSearch

Input data:
  an initial guess of the step parameter r_0
  the maximum step value r_M
  the function $q(r)$
Input parameters:
  the slope parameter p for the roof function $\lambda(r)$;
Output:
  an acceptable step parameter r

Initialize:
  $r \leftarrow r_0$;
While $r < r_M$ **and** $q(r) < \lambda(r)$,
  $r \leftarrow 2 \cdot r$; ....{4.1}
While $r < r_M$ **or** $q(r) > \lambda(r)$,
  $r \leftarrow r/2$;.................................{4.2}
Return r.

Figure 4. Line search procedure for MDS-PD

segment as r, such that $r \le r_m < 2 \cdot r$, whence $r > r_m/2$. In other words, the returned value is always in the upper half of the interval $[0, r_m]$ and we accept this as sufficient progress toward the solution, thus eliminating some more computationally demanding progress criteria that would require calculation of $q'(r)$ at points other than $r = 0$ or cannot always return a nonzero r (cf. Nocedal and Wright 1999; Frandsen et al. 2004).

It remains to show how to calculate the slope of $\lambda(r)$, that is $pq'(0)$ (Eq. 12). Given a configuration $\mathbf{z}$ and a direction $-\mathbf{g} = -\nabla E(\mathbf{z}, \boldsymbol{\Delta}, \mathbf{W}, \mathbf{I})$, the configuration $\mathbf{z}'$ as a function of r (7) can be conveniently represented as a column-vector function

$$\mathbf{M}(-r\mathbf{g}, \mathbf{z}) \tag{14}$$

whose j-th entry is the Möbius transform

$$M_j(r) = \frac{-rg_j + z_j}{-rg_j \overline{z_j} + 1}.$$

The associated embedding error as a function of r is then

$$q(r) = E(\mathbf{M}(-r\mathbf{g}, \mathbf{z}), \boldsymbol{\Delta}, \mathbf{W}, \mathbf{I}), \tag{15}$$

and it can be easily shown that its slope is given by

$$q'(r) = \frac{d}{dr}q(r) =$$
$$= (Re\,\mathbf{M}'(-r\mathbf{g}, \mathbf{z}))^T Re\,\nabla E(\mathbf{M}(-r\mathbf{g}, \mathbf{z}), \boldsymbol{\Delta}, \mathbf{W}, \mathbf{I})$$
$$+ (Im\,\mathbf{M}'(-r\mathbf{g}, \mathbf{z}))^T Im\,\nabla E(\mathbf{M}(-r\mathbf{g}, \mathbf{z}), \boldsymbol{\Delta}, \mathbf{W}, \mathbf{I})$$

where the entries of $\mathbf{M}'(-r\mathbf{g}, \mathbf{z})$ are given by

$$M_j'(r) = \frac{d}{dr}M_j(r) = g_j \frac{|z_j|^2 - 1}{(1 - rg_j \overline{z_j})^2}.$$

We thus have a general explicit formula for calculating $q'(r)$ given a configuration $\mathbf{z}$ and the corresponding gradient $\mathbf{g}$ of E at $\mathbf{z}$. In particular, this formula can be used to calculate $pq'(0)$, the slope of $\lambda(r)$.

III. MULTIDIMENSIONAL SCALING IN THE PD

A. Objective Functions and Gradients

The iterative minimization method presented in Sec. II requires a choice of an embedding error function with continuous first derivatives. In this work we consider the least squares error function

$$E = c \sum_{j=1}^{n} \sum_{k=j+1}^{n} c_{jk} (d_{jk} - a\delta_{jk})^2. \tag{16}$$

We note that (16) is a general form from which several special embedding error functions can be obtained by substituting appropriate values of the constants c, c_{jk}, and a. Examples include:

- Absolute Differences Squared (ADS)

$$E = \sum_{j=1}^{n} \sum_{k=j+1}^{n} w_{jk} (I_{jk}(d_{jk} - a\delta_{jk}))^2 \tag{17}$$

- Relative Differences Squared (RDS)

$$E = \sum_{j=1}^{n} \sum_{k=j+1}^{n} w_{jk} \left(I_{jk} \frac{d_{jk} - a\delta_{jk}}{a\delta_{jk}} \right)^2 \tag{18}$$

- Sammon Stress Criterion (SAM)

$$E = \frac{1}{a\sum_{j=1}^{n} \sum_{k=j+1}^{n} I_{jk}\delta_{jk}} \cdot \sum_{j=1}^{n} \sum_{k=j+1}^{n} w_{jk} \frac{(I_{jk}(d_{jk} - a\delta_{jk}))^2}{a\delta_{jk}} \tag{19}$$

As the most general case of (16), individual importance dependent on the input dissimilarities can be assigned to the pairwise error terms using the the weights terms w_{jk}.

MDS-PD also requires calculation of the gradient of the error function. For a general error function, closed form symbolic derivatives may or may not exist, and in the latter case one can resort to approximating the gradient using finite difference calculations. Numerical approximation may also incur lower computational cost than the formal derivatives. However, the use of numerical derivatives can introduce additional convergence problems due to limited machine precision.

For the sum (16), both approaches are possible. A symbolic derivation of the gradient of (16), including both the Euclidean and hyperbolic cases, can be easily carried out and is omitted here for brevity. From the obtained result, symbolic derivatives of (17)–(19), as well as any other special cases derivable from (16) can be obtained by substituting appropriate constants.

B. Local vs. Global Minima

MDS-PD, being a steepest descent method that terminates at near-zero progress, can find a *stationary point* of the objective function. In the least squares case, if the value at the returned solution is close to zero (that is, $E < \varepsilon_E$), then the final configuration can be considered a global minimizer that embeds the input metric with no error. In all other cases, a single run of MDS-PD cannot distinguish between local and global points of minimum or between a minimizer and a stationary point. A traditional way of getting closer to the global minimum in MDS is to run the minimization multiple times with different starting configurations. Expectedly, there will be accumulation of the results at several values, and the more values are accumulated at the lowest accumulation point, the better the confidence that the minimal value represents a global minimum i.e. the least achievable embedding error.

C. Input Data Curvature Matching

The objective functions used in metric Euclidean MDS are typically constructed to be *scale-invariant* in the sense that scaling the input dissimilarities and the coordinates of the output configuration with the same constant factor a does not change the embedding error. This is possible for Euclidean space since the Euclidean distance function scales by the same constant factor as the point coordinates: $(\sum_{s=1}^{L}(a \cdot y_{js} - a \cdot y_{ks})^2)^{1/2} = a \cdot d_{jk}$. Thus, for example, if d_{jk} is the Euclidean distance, then the sums (18) and (19) are scale-invariant, whereas (17) is not. However, when d_{jk} is the hyperbolic distance function (2), none of the (17)–(19) is scale-invariant. Therefore, the simplest ADS error function (17) may be a preferable choice for reducing the computational cost in the hyperbolic case. Nevertheless, for our numerical experiments we choose to apply the Sammon criterion (19) so as to facilitate numerical comparison between the final embedding errors for the Sammon map in the Euclidean plane and MDS-PD.

The lack of scale-invariance of the hyperbolic distance formula (2) implies an additional degree of freedom in the optimization of the embedding error – the *dissimilarity scaling factor*. In Eqs. (16)–(19) this extra degree of freedom is captured via the parameter a that scales the original entries of the dissimilarity matrix. The dependency of the embedding error of MDS-PD on the dissimilarity scaling factor a varies with the type of input data and is investigated in more detail in Section IV.

IV. MDS-PD: EXPERIMENTAL RESULTS

Following the specification in the previous sections, we successfully implemented MDS-PD with the error functions (17)–(19). In this section we show a few illustrative results of our experimental study using MDS-PD on synthetic as well as real-world data. Some of the methods we used to verify the correctness of our specification are also discussed below.

A. An Illustrative Example

As a first example, we used a random configuration of seven points in the Poincaré disk. We populated the input dissimilarity matrix with the hyperbolic inter-point distances and started MDS-PD from another randomly-generated seven point initial configuration in the PD. Fig. 5 shows the trajectories traveled by the points during the minimization. The clear points denote the initial configuration and the solid points represent the final point configuration. Fig. 6 shows the MDS-PD internal parameters vs. the iteration number for this example: In Fig. 6a, the embedding error E monotonically decreases with every iteration; the iterations terminated with the $E < \varepsilon_E = 10^{-6}$ condition, which means that likely the output configuration represents the global minimum and the final inter-point distances match the input dissimilarities very closely. The step-size parameter r was initialized with a value of 1 and assumed only values of the form 2^k, for integral k (Fig. 6b). The exponential character of the change of r according as {4.1} and {4.2} (Fig. 4) ensures the low computational cost of the line search subprogram and in our numerical studies proved superior to other line-search strategies, including exact search or adaptive approximate step-size parameter. The refining of the step size as the current configuration approaches a local minimum of the error function, on the other hand, is achieved by the decrease of the gradient norm. This is further illustrated in Figs. 6c and 6d.
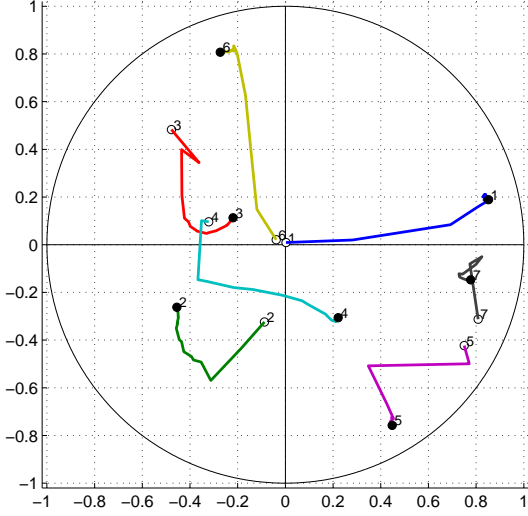
Figure 5. The minimization trajectory for a seven point configuration using MDS-PD. The clear and the solid points are respectively the initial and the final point configuration.
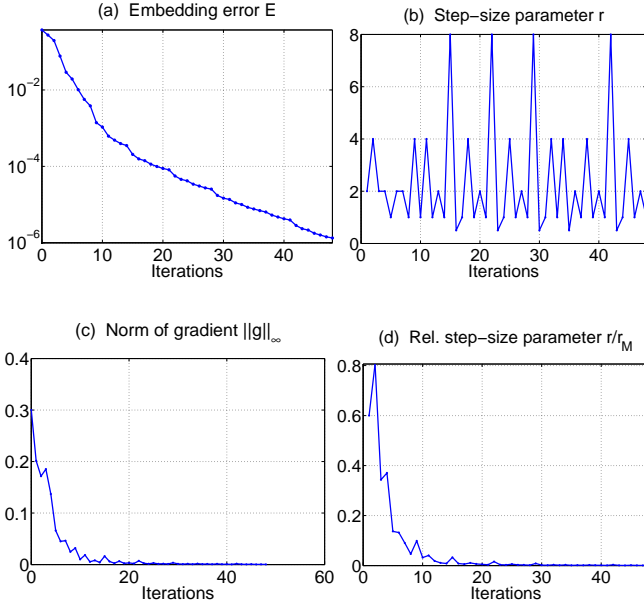

Figure 6. The MDS-PD internal parameters vs. the iteration number for the seven point example of Fig. 5: (a) the embedding error E, (b) the step-size parameter r, (c) the norm of the gradient $\|\mathbf{g}\|_\infty$, and (d) the step-size parameter relative to the maximum allowed value r/r_M.

B. Scaling of Synthetic Data

To investigate the dependency of the embedding error on the dissimilarity scaling factor a, we used as input the inter-point distances obtained from random sets of points residing on surfaces of constant positive, zero or negative curvature (i.e. respectively a sphere, a Euclidean plane and the Poincaré disk model of the hyperbolic plane.) The corresponding distances (spherical, Euclidean and hyperbolic) for all pairs were used as dissimilarities

in this experiment. The embedding error function was the Sammon criterion (19). We also used noisy inputs obtained by replacing each original dissimilarity δ_{jk} with a random number uniformly distributed in the interval $\left[(1-e_m)\delta_{jk}, (1+e_m)\delta_{jk}\right]$ for a chosen noise level $e_m < 1$.

Fig. 7 shows the typical effects of dissimilarity scaling for several Euclidean, spherical, and hyperbolic graphs. Cases (a), (c), and (e) illustrate the variation of the embedding error for noiseless input data, with the number of points as a parameter (20 and 60 points.) Cases (b), (d), and (f) illustrate the variation of the embedding error for noisy input data and are parametrized by the amount of measurement noise ($e_m = 0, 10, 20, 30\%$.) Each point in the diagrams (a)–(f) was obtained as a minimum Sammon stress in a series of 70 replicates of MDS-PD for different randomly chosen initial configuration in the PD. The smoothness of the obtained curves demonstrates that for the chosen problems, this number of replicates was enough to approach the global minimum achievable embedding error for each simulated value of a. The results are drawn on semilogarithmic axes in order to show more details toward small a values.

Locally, the Poincaré disk model, distance-wise "looks like" the Euclidean plane scaled with some constant factor. For example, in the vicinity of a point z_0, the hyperbolic distance formula (2) becomes $d_{\mathbb{D}}(z_j, z_k) \approx |z_j - z_k| \cdot 2/(1 - |z_0|^2)$. Therefore, for a sufficiently small scaling factor a and sufficiently many replicates, metric MDS implementations for the PD model and for the Euclidean plane using the same scale-invariant (for Euclidean distances) error function, should return approximately equal embedding errors for the final configurations. (A sufficiently small value of a is one that would make the final configuration land in a sufficiently small neighborhood of a point in the PD.) In this sense, MDS-PD is a generalization of an Euclidean MDS algorithm. We used these observations to verify that our MDS-PD implementation returned the expected error values for small scaling factors. Indeed, as Fig. 7 shows, for small a values, the Euclidean graphs were embeddable with no error, and the other two graph types had stress that numerically matched the output of other available Euclidean MDS implementations using the Sammon stress criterion.

In the cases (e) and (f), the original configurations are residing on a hyperbolic plane, and therefore are embeddable with zero stress in the PD model for some value of a ($a = 1$ in this synthetic example). For this
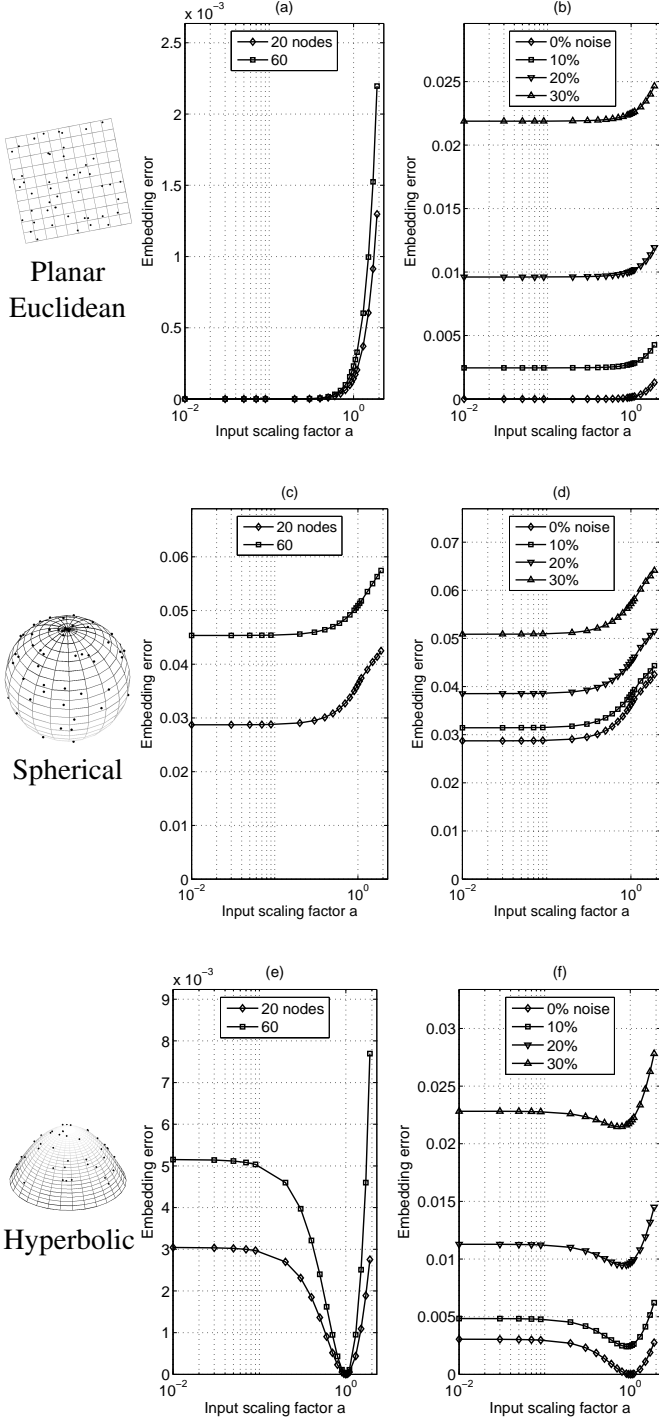
Figure 7. Effects of dissimilarity scaling for Euclidean, spherical, and hyperbolic graphs. Figures (a), (c), and (e) show the variation of the embedding error for noiseless input data, with the number of nodes as a parameter. Figures (b), (d), and (f) show the variation of the embedding error for noisy input data parametrized by the amount of measurement noise.

value, our implementation of MDS-PD was able to find the original configuration up to hyperbolic-distance preserving Möbius transformations. The existence of graphs with a hyperbolic "underlying structure" that are embeddable with notably lower stress in 2-dimensional hyperbolic than in 2-dimensional Euclidean space is an important illustration of the usefulness of MDS in the Poincaré disk. Candidate graphs having such hyperbolic-like structure are real-world communication or social networks that tend to have a strongly interconnected core and a sparser periphery of tendrils. MDS-PD can be used in such contexts to investigate the "hyperbolicity" of the input data and arrive at lower stress dissimilarity embedding. The diagrams (b), (d), and (f) (Fig. 7) also demonstrate that relatively high noise levels in the measured data do not significantly change the suitability for embedding in the PD in the cases when the original dissimilarity matrix has a natural underlying 2-dimensional space.

C. Real-World Graph Examples

To further demonstrate the ability of the Poincaré disk to accommodate lower stress 2-dimensional embeddings than classical Euclidean MDS for certain graph types, we applied MDS-PD to dissimilarity matrices obtained from several real-world datasets. In this section we summarize the results.

1) The Iris Dataset: As a first experiment, we applied MDS-PD to the Iris dataset (Anderson 1935). This classical dataset consists of 150 4-dimensional points from which we extracted the Euclidean inter-point distances and used them as input dissimilarities. The embedding error as a function of the scaling factor a is shown in Fig. 8. Each value in the diagram is obtained as a minimum embedding error in a series of 100 replicates starting from randomly chosen initial configurations. Minimal embedding error overall was achieved for $a \approx 4$. The improvement with respect to the 2-dimensional Euclidean case was 10%. The Iris dataset is an example of dimensionality reduction of an original higher-dimensional dataset that can be done more successfully using the PD model.

2) Political Books: An interesting network was presented by Krebs (2008), who assembled a connectivity graph of political books frequently bought together during an election campaign. In the graph version we used, there were 105 nodes representing books and a total of 441 undirected, unweighted links between books that

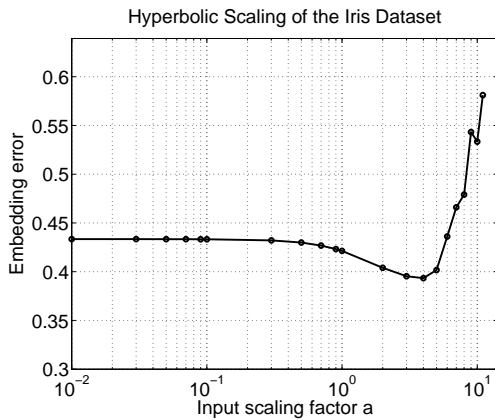Hyperbolic Scaling of the Iris Dataset

Figure 8. The effect of scaling of the dissimilarities on the embedding error for the Iris Dataset (Anderson 1935). The input dissimilarities are the Euclidean distances between pairs of original points. This MDS-PD result reveals that the Iris dataset is better suited for embedding to the hyperbolic plane that to the Euclidean plane.

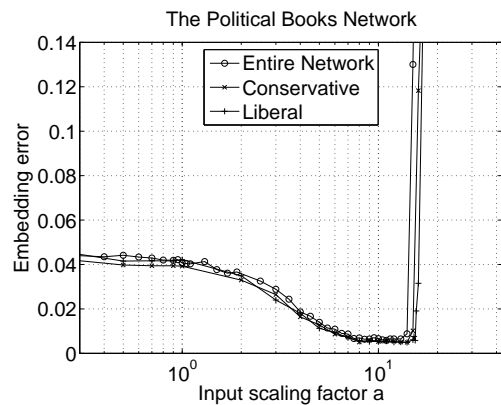The Political Books Network

Figure 9. The embedding error as a function of the input scaling factor for the network of political books (Krebs 2008). The input dissimilarities are simply indicators of the presence or absence of links between the network nodes. The political books network is an example of unweighted, undirected real-world graph data that can be embedded with lower error in the PD model than in the Euclidean plane.
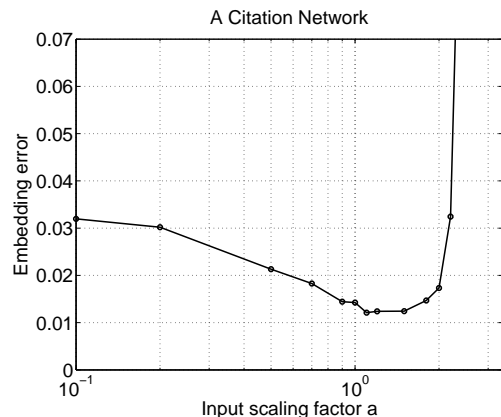
A Citation Network

Figure 10. The embedding error as a function of the input scaling factor for the network of citations (Newman 2001). The network is an example of weighted graph data that can be encoded with lower embedding error in the PD model than in the Euclidean plane. The input dissimilarities are capturing the strength of the collaboration between pairs of authors.

were frequently bought together. We obtained dissimilarities by assigning self-dissimilarity of 0, dissimilarity of 1 for co-purchased books and a missing (unknown) dissimilarity for the remaining pairs and applied MDS-PD to the resulting dissimilarity matrix. We also conducted the experiment using only the liberal and the conservative subgraphs (43 and 49 points respectively). The obtained minimum embedding error of 150 replicates as a function of the scaling factor a is shown in Fig. 9. We note that there were remarkable gains of using the PD model instead of the Euclidean plane. For the overall graph, the minimal stress was 7.6 times smaller than the Euclidean stress. The liberal and conservative components alone achieved improvement of 8.8 and 9 times with respect to the Euclidean case.

3) Citation Network: The citation network that we used in this experiment was compiled by Newman (2001) from bibliographies of review articles on networking. We extracted the largest connected component from the graph which consisted of 379 nodes representing authors. There were 244 edges in the graph with weights s_{jk} representing the strength of the collaborative ties. We have calculated dissimilarities from these data using $\delta_{jk} = \text{const} - s_{jk}$ and applied the MDS-PD algorithm. The obtained minimum embedding error of 50 replicates as a function of the scaling factor a is shown in Fig. 10. The overall minimum embedding error was 2.63 times lower than the stress obtained using Euclidean MDS.

V. RELATED WORK

A multidimensional scaling algorithm for fitting distances to constant-curvature Riemannian spaces was given by Lindman and Caelli (1978). This work uses the hyperboloid model of the hyperbolic space which requires an $n+1$-dimensional Euclidean space to represent an n-dimensional hyperbolic space, and is less suitable for visualization purposes in the case of the hyperbolic plane.

The applicability of metric multidimensional scaling in the Poincaré disk model of the hyperbolic plane was studied by Walter (2004). The study focused on the task of embedding higher-dimensional point sets into 2-dimensional configurations for the purpose of interactive visualization. It was demonstrated that the PD has capacity to accommodate lower stress embedding
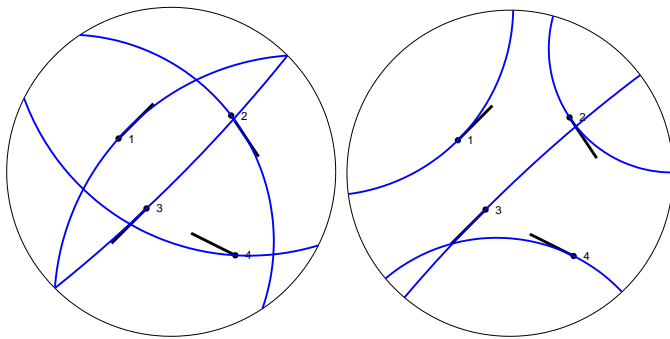
Figure 11. Comparison of the point trajectories: H-MDS of J. Walter Walter (2004) (left) vs. MDS-PD hyperbolic lines (right)

than the Euclidean plane. Several important pointers to the difficulties one encounters in implementing such algorithms were given, but a definite specification or implementation was not provided.

The proposed method to convert the seemingly constrained optimization problem to an unconstrained one (Walter 2004, Eq. 12) ensures that the moving configuration would stay inside the model during the optimization. However, this transformation fails to follow the distance realizing (hyperbolic) lines, or even Euclidean lines. The problem is illustrated in Fig. 11. The possibility that the dissimilarity matrix has missing values was also not addressed in this work, as the dissimilarities were generated from higher-dimensional points. Inputs derived from graph data, however, are typically sparse.

Shavitt and Tankel (2008) presented numerical examples illustrating that models of the hyperbolic space with circular symmetry may be better suited than Euclidean spaces for embedding network graphs with core-and-tendrils structure. Their work concentrated on applications specific to communication networks where dissimilarities for each pair of points are derived from the lengths of the shortest paths in the graph. For such applications, the authors' insight about the choice of the Poincaré disk model was that the shortest paths in the studied networks often pass through the core and are therefore longer than the straight-line distance, and this observation empirically matches the behavior of distance function in the chosen model. In order to avoid the constrained nature of the coordinates in the PD, the authors eventually resorted to the hyperboloid model of the hyperbolic plane, omitting the details.

The "big-bang simulation" (BBS) numerical method used in (Shavitt and Tankel 2008), is discussed in (Shavitt and Tankel 2004). BBS is a variant of a steepest descent method that models the point configuration as an inertial system in a force-generating field. Termination is guaranteed by introducing empirical dampening in the mechanical system. The initial configuration in BBS is always chosen to be a single point in which all particles are collocated, ensuring a fair initial amount of potential energy. Another heuristic feature of BBS is that the objective function changes several times during the minimization in a way that increases the error sensitivity of the penalty terms. The particle inertia in BBS in conjunction with a stepwise changing objective function possibly allows the method to escape a few local minima before termination. However, the advantages of these heuristics in avoiding local minima, compared to a computationally simpler, single phase minimization procedure, were not clearly demonstrated. It is conceivable that the inertial minimum-avoiding mechanism, which comes at an increased computational cost, may as well cause the configuration to leave the global minimum, or a lower local minimum before stopping in a higher one. Finally, since BBS can only be started from one possible initial configuration, it has a deterministic outcome once the heuristic parameters such as friction and time slice are chosen; with this choice, the possibility that the final result is improved by restarting from different initial conditions, is eliminated.

Numerous methods that are more likely to find a lower minimum than the simplest repeated descent methods in a single run have been contemplated in the numerical optimization literature. However, to guarantee in general that the global minimizer is found is difficult with any such method. It may be necessary to resort to running the sophisticated methods several times as well in order to gain confidence in the final result. Since these methods are usually computationally more complex or incorporate a larger number of heuristic parameters, the incurred computational and implementational costs often offset the benefits of their sophistication.

VI. CONCLUSION

We developed the details of MDS-PD, an iterative minimization method for metric multidimensional scaling of dissimilarity data in the Poincaré disk model of the hyperbolic plane. While our exposition concentrated on a simple steepest descent minimization with approximate binary hyperbolic line search, we believe that elements of the presented material will also be useful as a general recipe for transferring other, more sophisticated iterative methods of unconstrained optimization to various models of the hyperbolic space.

We believe that the algorithm specification herein allows for easy implementation and represents a useful novel tool for the scientific community and that its application will enable a wide range of scientific inquiry spanning beyond the area of numerical optimization methods.

Our initial numerical experiments using both synthetic and real-world data suggest that MDS-PD achieves significantly better results compared to its Euclidean counterpart for naturally arising networks of interaction. This behavior may ultimately be attributed to the less-restricted axiomatic foundation of the hyperbolic geometry compared to its Euclidean counterpart. Our results justify the development effort and encourage future work in the directions of generalizing this algorithm to higher-dimensional models of the hyperbolic space, improving its efficiency, and establishing novel uses and applications.

VII. ACKNOWLEDGMENTS

REFERENCES

E. Anderson. The irises of the Gaspé peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

James W. Anderson. *Hyperbolic Geometry*. Springer, 2nd edition, 2007.

P. M. Bentler and David G. Weeks. Restricted multidimensional scaling models. *J. Math. Psychol.*, 17:138–151, 1978.

Bruce Bloxom. Constrained multidimensional scaling in N-spaces. *Psychometrika*, 43:397–408, 1978.

Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*. Springer, Berlin, 2nd edition, 2005.

J. D. Carroll and P. Arabie. Multidimensional scaling. *Ann. Rev. Psychol.*, 31:607–649, 1980.

T. F. Cox and M. A. A. Cox. Multidimensional scaling on a sphere. *Communications in Statistics*, 20:2943–2953, 1991.

Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling (Monographs on Statistics and Applied Probability)*. Chapman & Hall/CRC, 2nd edition, 2000.

Andrej Cvetkovski and Mark Crovella. Hyperbolic embedding and routing for dynamic graphs. In *Proceedings of IEEE Infocom 2009*, pages 1647–1655, Apr 2009.

J. De Leeuw and W. Heiser. Theory of multidimensional scaling. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of statistics*, volume 2. North-Holland, 1982.

P. E. Frandsen, K. Jonasson, H. B. Nielsen, and O. Tingleff. *Unconstrained Optimization*. IMM, DTU, 3rd edition, 2004.

Robert Kleinberg. Geographic routing using hyperbolic space. In *Proceedings of IEEE Infocom 2007*, pages 1902–1909, May 2007.

V. Krebs. Case studies - Political Book Networks, 2008. Available online at www.orgnet.com.

D. Krioukov, F. Papadopoulos, M. Boguna, and A. Vahdat. Efficient navigation in scale-free networks embedded in hyperbolic metric spaces. In arXiv: Article-id:0805.1266, 2008.

John Lamping and Ramana Rao. Laying out and visualizing large trees using a hyperbolic space. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 13–14, New York, NY, USA, 1994. ACM. ISBN 0-89791-657-3. doi: http://doi.acm.org/10.1145/192426.192430.

H. Lindman and T. Caelli. Constant curvature Riemannian scaling. *J. Math. Psychol.*, 2:89–109, 1978.

M. E. J. Newman. Scientific collaboration networks II. Shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64(1):016132, 1–7, Jun 2001. doi: 10.1103/PhysRevE.64.016132.

Jorge Nocedal and Stephen. J. Wright. *Numerical Optimization*. Springer, 1999.

J.W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, C-18(5):401–409, May 1969. ISSN 0018-9340.

Yuval Shavitt and Tomer Tankel. Big-bang simulation for embedding network distances in euclidean space. *IEEE/ACM Trans. Netw.*, 12(6):993–1006, 2004. ISSN 1063-6692.

Yuval Shavitt and Tomer Tankel. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Trans. Netw.*, 16(1):25–36, Feb. 2008. ISSN 1063-6692.

Jörg A. Walter. H-MDS: a new approach for interactive visualization with multidimensional scaling in the hyperbolic space. *Information Systems*, 29(4):273 – 292, 2004. ISSN 0306-4379.