Computational Statistics and Data Analysis 104 (2016) 51–65

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Visualizing the effects of a changing distance on data using continuous embeddings

Gina Gruenhage [a,b,*], Manfred Opper [a], Simon Barthelme [c]

[a] *Department of Computer Science, Technische Universität Berlin, Berlin, Germany*
[b] *BCCN Berlin, Berlin, Germany*
[c] *CNRS, GIPSA-lab, 11 Rue des Mathématiques, 38400 Saint-Martin-d'Hères, France*

HIGHLIGHTS

- A continuous version of Multidimensional Scaling is presented (cMDS).
- The algorithm is designed based on CCCP and allows for easy variation.
- It is noted that most data analyses are based on a specific distance function.
- The method visualizes the data with respect to various possible distance functions.
- An R-package (cmdsr) is provided to facilitate the use of the cMDS.

ARTICLE INFO

Article history:
Received 25 November 2015
Received in revised form 17 June 2016
Accepted 17 June 2016
Available online 27 June 2016

Keywords:
Dimensionality reduction
Multidimensional scaling
Visualization
Data exploration

ABSTRACT

Most Machine Learning (ML) methods, from clustering to classification, rely on a distance function to describe relationships between datapoints. For complex datasets it is hard to avoid making some arbitrary choices when defining a distance function. To compare images, one must choose a spatial scale, for signals, a temporal scale. The right scale is hard to pin down and it is preferable when results do not depend too tightly on the exact value one picked. Topological data analysis seeks to address this issue by focusing on the notion of neighborhood instead of distance. It is shown that in some cases a simpler solution is available. It can be checked how strongly distance relationships depend on a hyperparameter using dimensionality reduction. A variant of dynamical multi-dimensional scaling (MDS) is formulated, which embeds datapoints as curves. The resulting algorithm is based on the Concave–Convex Procedure (CCCP) and provides a simple and efficient way of visualizing changes and invariances in distance patterns as a hyperparameter is varied. A variant to analyze the dependence on multiple hyperparameters is also presented. A cMDS algorithm that is straightforward to implement, use and extend is provided. To illustrate the possibilities of cMDS, cMDS is applied to several real-world datasets.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

The notion of distance is at the core of data analysis, pattern recognition and machine learning: most methods need to know how similar two datapoints are. The choice of distance metric is often a hidden assumption in algorithms. For complex

* Correspondence to: TU Berlin, Fakultät IV, Elektrotechnik und Informatik, Sekr. MAR 4-2, Marchstrasse 23, D-10587 Berlin, Germany.
E-mail addresses: gina.gruenhage@bccn-berlin.de (G. Gruenhage), manfred.opper@tu-berlin.de (M. Opper), simon.barthelme@gipsa-lab.fr (S. Barthelme).

http://dx.doi.org/10.1016/j.csda.2016.06.006
0167-9473/© 2016 Elsevier B.V. All rights reserved.

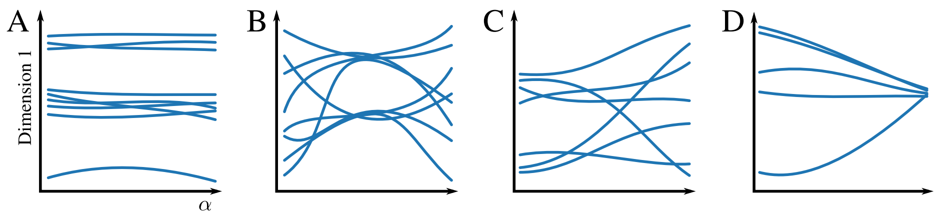

Fig. 1. Sketches of different effects on the data structure that emerge when varying a hyperparameter in a distance function. The x-axis shows the hyperparameter α, the y-axis is the embedding dimension. (A) Invariance: patterns hold independent of the hyperparameter. (B/C) Structure emerges only for certain values of the hyperparameter. (C) Declustering: clusters are lost with increasing hyperparameter. (D) Information loss: Structure collapses with increasing hyperparameter.

data, distance or similarity is not uniquely defined. On the contrary, they can be arbitrary to some extent (Carlsson, 2009). It is, for example, often possible to describe signals on different temporal or spatial scales, and distance functions will give a certain scale more weight than another. Each datapoint might describe several features, and there is often no unique, optimal way to weigh the features when computing a distance measure: are two individuals more alike if they have similar eye color or hair color, or do we think the shape of the nose matters most?

There are ways around that problem. One is to select the distance function that is best adapted to the task at hand, for example the one that gives the best performance in classification (this is effectively what is done in kernel hyperparameter selection Schölkopf and Smola, 2002). Another is to give up on distance and rely instead on the weaker notion of neighborhood (Lum et al., 2013).

We argue here that a third option is available. One may study how the shape of the data evolves under a change in the distance metric by representing the data in lower dimension. We suppose that a family of distance functions $d_\alpha(x, y)$ is defined by varying a hyperparameter $\alpha \in [0, 1]$, where α can represent, for example, different scales or the mixing proportion of features. Please note that α does not have to be defined on this interval, but it seems natural to start with a setting that is familiar from, e.g., convex combinations. Suppose that for a given level of α the relative distances between datapoints are well described by representing the datapoints as points on the line. As we vary α the points will move, so that each point now describes a curve. Many scenarios are possible, and we sketch them in Fig. 1. We may have full or partial *invariance*: patterns in the data that hold regardless of the value of the hyperparameter (Fig. 1(A)). On the other hand, the structure in the data may appear only for certain values of α (intermediate values in Fig. 1(B) and rather small values in C), indicating that these values are more useful than others for characterizing the data. Analyzing the evolution of structures in the data might reveal interesting dependencies, for example, declustering (Fig. 1(C)) or loss of information (Fig. 1(D)).

To visualize the effects of varying the distance function we suggest to embed data into a space of smooth curves, forming what we call continuous embeddings: in continuous embeddings each datapoint is embedded as a smooth curve in $\mathbb{R}^d$. We will show that this approach is quite general.

Our implementation of continuous embeddings is based on multi-dimensional scaling (MDS), one of the most widely-used tools for dimensionality reduction (Buja and Swayne, 2002; Buja et al., 2008). MDS builds on the pairwise relation between single data points and has an intuitive way of characterizing the structure in high-dimensional data. MDS supposes that one has distance information available, that is, we can characterize the data by a distance matrix. MDS seeks to find a set of points in a low dimensional Euclidean space, such that the Euclidean distances between points approximate the original distances. An exception is *spherical* MDS, where the embedding is constrained to a spherical manifold. MDS goes back to the 1950s, when it was first introduced as classical scaling (Torgerson, 1952). In classical scaling, the distance matrix is transformed to a matrix of inner products from which an embedding can be computed using eigendecompositions (Torgerson, 1952, 1958; Gower, 1966). Classical scaling finds a perfect embedding when the data can indeed be embedded exactly, but in all realistic cases distance matrices are not exactly Euclidean and *distance* scaling is more appropriate. Kruskal (1964) introduced distance scaling by defining a cost function, *Stress*, that directly measures the error between original and embedding distances. This cost function is then optimized over the space of embedding matrices which can be done using gradient descent. Since the early work on MDS many other variants and optimization solutions have been discussed. So called non-metric variants of MDS seek to only recover the ranks of distances (Shepard, 1962). Ramsay (1997, 1978b) introduces a statistical model for MDS, allowing for a maximum likelihood estimate. This approach is implemented in *Multiscale* (Ramsay, 1978b). Other MDS variants based on Stress include Sammon's mapping (Sammon, 1969), elastic stress (McGee, 1966), multidimensional unfolding (Borg and Groenen, 2005) and local MDS (Chen and Buja, 2009). Isomap (Tenenbaum et al., 2000) is also related to MDS. Here, distances are computed as geodesic distances on a manifold, which are then embedded with classical scaling. In terms of optimization one of the most popular approaches is SMACOF, a majorization method for MDS (Guttman, 1968; De Leeuw, 1977; De Leeuw and Heiser, 1977; De Leeuw, 1988).

Here, we introduce a continuous version of MDS (cMDS) by adding a smoothing penalty to the MDS cost function. Similar ideas have been used in the visualization of dynamic networks. A network is commonly represented as a graph. A 2D embedding of a static graph is often constructed using MDS or similar methods (Kamada and Kawai, 1989; Gansner et al., 2005). In the dynamical context, where a graph is measured over time, it is important to preserve the so-called "mental map" when jumping from one timepoint to the next (Misue et al., 1995). Early work on such *controlled stability* was done

Download English Version:

https://daneshyari.com/en/article/6868931

Download Persian Version:

https://daneshyari.com/article/6868931

Daneshyari.com