Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

Optimal Axes for Data Value Estimation in Star Coordinates and Radial Axes Plots

M. Rubio-Sánchez[1] and D. J. Lehmann[2] and A. Sanchez[1] and J. L. Rojo-Álvarez[1]

[1]Universidad Rey Juan Carlos, Madrid, Spain
[2]University of Magdeburg & Ostfalia University of Applied Sciences Wolfenbüttel, Germany

Abstract

Radial axes plots are projection methods that represent high-dimensional data samples as points on a two-dimensional plane. These techniques define mappings through a set of axis vectors, each associated with a data variable, which users can manipulate interactively to create different plots and analyze data from multiple points of view. However, updating the direction and length of an axis vector is far from trivial. Users must consider the data analysis task, domain knowledge, the directions in which values should increase, the relative importance of each variable, or the correlations between variables, among other factors. Another issue is the difficulty to approximate high-dimensional data values in the two-dimensional visualizations, which can hamper searching for data with particular characteristics, analyzing the most common data values in clusters, inspecting outliers, etc. In this paper we present and analyze several optimization approaches for enhancing radial axes plots regarding their ability to represent high-dimensional data values. The techniques can be used not only to approximate data values with greater accuracy, but also to guide users when updating axis vectors or extending visualizations with new variables, since they can reveal poor choices of axis vectors. The optimal axes can also be included in nonlinear plots. In particular, we show how they can be used within RadViz to assess the quality of a variable ordering. The in-depth analysis carried out is useful for visualization designers developing radial axes techniques, or planning to incorporate axes into other visualization methods.

CCS Concepts

• ***Human-centered computing*** → ***Visualization techniques;*** *Visualization theory, concepts and paradigms;* • ***Mathematics of computing*** → *Exploratory data analysis;*

1. Introduction

Plots based on radial axes [Gab71, Kan00, LT13, RSSL17, SSRMJ*18] are multivariate visualization techniques that have been used mainly for exploratory purposes including cluster analysis, outlier and trend detection, or decision support tasks. In a broad sense, they can be understood as dimensionality reduction techniques, since they map high-dimensional numerical data points onto a lower-dimensional observable space, which is typically a plane. By visualizing the projected points through dots or other markers, users can obtain insights about the relationships between the data points, albeit the loss of information associated with the dimensionality reduction process.

There are several notable differences between visualization methods based on radial axes and most dimensionality reduction approaches (see [MH08, TLZM16, MHSG18, EMK*19, Sau20]). While the former are usually linear, the latter generally define nonlinear transformations. Most importantly, plots based on radial axes show information related to data attributes by depicting a set of "axis" vectors, where each one is associated with a different variable. Some methods also show straight lines that represent axes in the directions of the vectors. Unlike in other dimensionality reduction methods, the possibility to visualize these vectors and axis lines allows analysts to intuitively infer relationships between variables, and between the data points and the variables. Lastly, several methods based on radial axes are interactive in the sense that users can update the axis vectors freely, creating customized mappings for examining the data from multiple points of view.

One of the main challenges users face when working with radial axes methods is deciding where to locate the axis vectors. Updating the direction and length of an axis vector, or choosing the coordinates of a new one to be added to the visualizations, is far from trivial. Users must consider the data analysis task, domain knowledge, the relative importance of each variable, or the correlations between variables, among other factors. Another issue is the difficulty to approximate and compare high-dimensional values in the embedding space, which can negatively affect searching for data with particular characteristics, analyzing the most common data values in clusters, inspecting outliers, etc.

In this paper we present and analyze several optimization approaches for finding axis vectors and lines that better reflect the

directions in which variable values are ordered or increase in the plots. Specifically, we focus on enhancing the ability of the radial methods to represent high-dimensional data values. The techniques can be used not only to approximate data values with greater accuracy, but also to guide users when updating axis vectors or extending visualizations with new variables, since they can reveal poor choices of axis vectors. The optimal axes can even be included in some nonlinear embeddings. In particular, we show how they can be used within RadViz [HGM*97, GJH*01, SGM08, DGRG12] to assess the quality of a variable ordering. While the paper proposes several algorithms based on optimization problems, it can also be considered as a theoretical work (we provide proofs and derivations in the lengthy supplemental material). The in-depth analysis carried out is useful for visualization designers developing radial axes methods, or planning to incorporate axes into other visualization techniques.

The paper is organized as follows. Section 2 reviews key concepts, introduces the main notation, and describes related work. In Section 3 we present the approaches for obtaining optimal axes in radial axes plots. Finally, Sec. 4 presents a discussion.

2. Related work and notation

Multivariate embeddings map each data sample $\mathbf{x} \in \mathbb{R}^n$ onto an embedded point $\mathbf{p} \in \mathbb{R}^m$. In this paper we will assume $m = 2$. Given a data set of cardinality N, $\mathbf{X}$ will represent the $N \times n$ data matrix whose rows contain the data samples, while $\mathbf{P}$ will be an $N \times 2$ matrix whose rows consist of the data samples' low-dimensional representations.

Methods based on radial axes define their mappings through a set of two-dimensional "axis" vectors $\mathbf{v}_i$, for $i = 1,\dots,n$, generally depicted with a common origin point, where $\mathbf{v}_i$ is associated with the i-th data variable (see Fig. 1). In this paper $\mathbf{V}$ will represent the $n \times 2$ matrix whose i-th row is $\mathbf{v}_i$. The main benefit of these methods resides in the possibility to visualize the axis vectors and their associated axis lines, which convey diverse information related to high-dimensional data values, correlations between variables, or the relevance of the variables in the visualizations.

2.1. Star coordinates

One of the earliest and most popular radial axes method is star coordinates (SC) [Kan00, Kan01]. Given a particular data sample $\mathbf{x}$, its associated low-dimensional embedded point $\mathbf{p}$ is:

$$\mathbf{p} = x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + \cdots + x_n\mathbf{v}_n = \mathbf{V}^\top\mathbf{x}. \tag{1}$$

Equivalently, in matrix notation the SC mapping is defined as:

$$\mathbf{P} = \mathbf{X}\mathbf{V}. \tag{2}$$

The method therefore produces a linear mapping of the data defined by $\mathbf{V}$, and explicitly shows its components through the axis vectors. Roughly speaking, $\mathbf{v}_i$ points towards a region in the plot where we would expect to find points with larger values for the i-th attribute, since increasing x_i shifts $\mathbf{p}$ in the direction of $\mathbf{v}_i$. In addition, the length of $\mathbf{v}_i$ is related to the relative contribution of the i-th variable to the visualization, assuming that the variables have a similar scaling. In practice the attributes are either normalized to lie in the [0,1] interval, or standardized (i.e., to have zero mean and unit variance).

There are two main ways to use the technique. On the one hand, the matrix of axis vectors $\mathbf{V}$ can be fixed. For example, it may be computed through automatic linear procedures such as principal component analysis (PCA) [Jol10], linear discriminant analysis (LDA) [McL04], optimal sets of projections [LT16b], general projective maps [LT16a], or many others. In these cases the axis vectors provide insight about the role of the variables in the automatic methods. On the other hand, users can specify and adapt the axis vectors interactively in order to generate desired linear mappings of the data, and search for data with particular characteristics, analyze cluster structure from multiple points of view, or detect outliers, among other tasks. Lastly, interaction in SC is effective for a moderate number of variables (up to 15-20) [ML19].

Figure 1(a) shows a SC plot of the Auto MPG data set, available at the UCI Machine Learning Repository [Lic13], for four of its variables. In this example we have chosen the axis vectors in order to characterize cars with high Horsepower and Acceleration, but low MPG, which appear at the top of the visualization. In addition, cars located towards the right will tend to have larger Displacement values. The graphic also includes an example of the method's linear combination through a concatenation of the lighter vectors (which are axis vectors scaled by the attribute values of a data sample) that forms a path that starts at the origin and ends at the embedded point. Note that there is usually an infinite number of paths that can end at an embedded point. Therefore, in practice it is very difficult to estimate the high-dimensional values (x_i) associated with an embedded point $\mathbf{p}$ by simply visualizing the plot, as shown in [RSS14].

2.2. Orthographic star coordinates

Orthographic star coordinates (OSC) [LT13] is a special case of SC where the columns of $\mathbf{V}$ are constrained to form an orthonormal set of vectors. In other words, $\mathbf{V}$ must be orthogonal: $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, where $\mathbf{I}$ is the 2×2 identity matrix. OSC enhances classical SC by avoiding linear distortions. For example, hyperspheres are transformed into circles, and not ellipses.

In practice it is cumbersome to select a set of axis vectors interactively that forms an orthogonal matrix. Note that when a user updates an axis vector (i.e., a row of $\mathbf{V}$) the rest need to be modified simultaneously. Therefore, users typically select some desired set of axis vectors, and subsequently replace them by the rows of an orthogonal matrix $\mathbf{V}_\perp$ with the same range as that of $\mathbf{V}$. The resulting matrix can be found in several ways. For example, it can be computed through a matrix multiplication:

$$\mathbf{V}_\perp = \mathbf{V}\mathbf{B}, \tag{3}$$

where $\mathbf{B}$ is an appropriate 2×2 nonsingular matrix. For instance, a well-known approach consists of computing the QR decomposition of $\mathbf{V}$, which is equivalent to performing the Gram-Schmidt orthonormalization procedure. In that case $\mathbf{V} = \mathbf{V}_\perp\mathbf{R}$, where $\mathbf{R}$ is an upper triangular 2×2 invertible matrix. Thus, $\mathbf{V}_\perp = \mathbf{V}\mathbf{R}^{-1}$. Figure 1(b) shows an OSC plot with standardized data.

A recent closely related variant is shape-preserving star coordinates (SPSC) [ML19]. It relaxes the orthogonality condition of OSC in order to scale the plots by some factor $\beta > 0$. The constraint in SPSC is therefore $\mathbf{V}^\top\mathbf{V} = \beta\mathbf{I}$, which can be beneficial when converting a SPSC plot into another one.

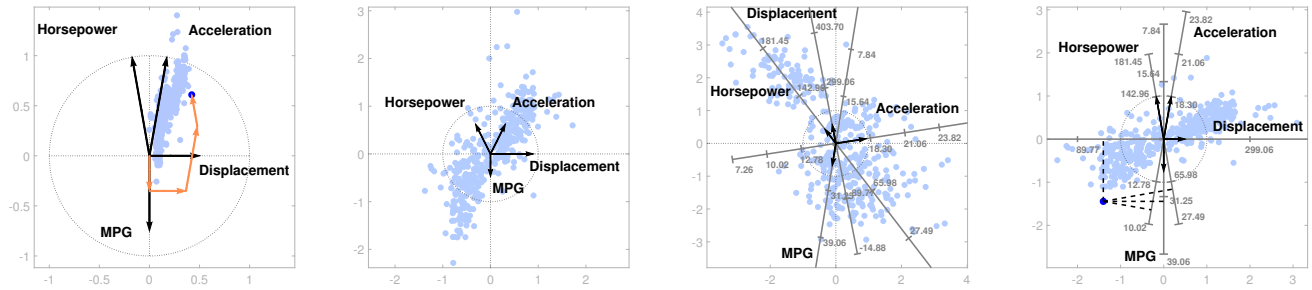

Figure 1: *Examples of plots based on radial axes of the Auto MPG data set. The dark axis vectors in each plot correspond to four variables (Horsepower, Acceleration, Displacement, and MPG) of the data set, while the dots represent individual samples (i.e., cars). In (a) the graphic is a SC plot, where we have included the concatenated orange arrows to symbolize the linear combination of (scaled) axis vectors that results in the darker embedded point. In this example we have chosen the axis vectors in order to show cars with high Acceleration and Horsepower, but low MPG, at the top of the plot. In addition, the cars can also be characterized horizontally according to their Displacement attribute. The graphic in (b) is an OSC plot, where the orthogonal transformation matrix is as similar as possible to the one used in (a). In (c) we show a PCA biplot with axis lines where the axis vectors are oriented very differently as in the previous plots. In (d) the graphic is an ARA plot with the same axis vectors as in (a). These last two visualizations also include labeled axes in order to help users approximate high-dimensional data attributes by projecting points orthogonally onto those axes. In this paper we focus on minimizing the sum of squared differences between these approximations and the true data values, which we denote as "estimation error". Finally, in (a) the data is normalized to lie in the [0,1] interval, while the rest of the visualizations use standardized (i.e., centered and of unit variance) data.*

2.3. Principal component biplots

Principal component biplots (PCB) [Gab71, GH95, Gre10, GGLlR11] are static linear plots that are optimal (in a least-squares sense) regarding the ability to represent high-dimensional data values on a lower-dimensional space. Specifically, PCB calculate sets of embedded points and axis vectors by solving the following optimization problem:

$$\underset{\mathbf{P}\in\mathbb{R}^{N\times 2},\mathbf{V}\in\mathbb{R}^{n\times 2}}{\text{minimize}} \quad \|\mathbf{P}\mathbf{V}^{\mathsf{T}}-\mathbf{X}_c\|_{\mathrm{F}}^2, \tag{4}$$

where $\mathbf{X}_c$ is a centered version of the data matrix $\mathbf{X}$, the subscript F denotes the Frobenius norm. Note that PCB solve for $\mathbf{P}$ and $\mathbf{V}$ simultaneously, and the matrix $\mathbf{P}\mathbf{V}^{\mathsf{T}}$ is an optimal approximation of the data. The solution is given by:

$$\mathbf{P}\mathbf{V}^{\mathsf{T}} = (k\mathbf{U}\mathbf{D}^{1-d})\left(\frac{1}{k}\mathbf{D}^{d}\mathbf{Z}^{\mathsf{T}}\right), \tag{5}$$

for some suitable (scaling) constants k and d, and where the product $\mathbf{U}\mathbf{D}\mathbf{Z}^{\mathsf{T}}$ is the (compact) singular value decomposition of the optimal rank 2 approximation of the data matrix $\mathbf{X}_c$, according to the (squared) Frobenius norm (see [EY36, RSSL17]). If $k=1$ and $d=0$ (i.e., $\mathbf{P}=\mathbf{U}\mathbf{D}$ and $\mathbf{V}=\mathbf{Z}$) the PCB is the well-known PCA plot, which can also be considered as an OSC plot since $\mathbf{V}^{\mathsf{T}}\mathbf{V}=\mathbf{I}$.

An important feature of biplots is the use of additional labeled (also denoted as "calibrated") axis lines that users can employ to estimate high-dimensional data values. Specifically, users extract these approximations ($\mathbf{P}\mathbf{V}^{\mathsf{T}}$) visually by projecting the embedded points orthogonally onto the axes. Note that biplots find optimal axis vectors and embedded points that minimize the squared differences between these approximations and the true data values, which we denote as "estimation errors". The methods presented in this paper will also minimize these estimation errors. Lastly, since the approximations are dot products between the embedded points and the axis vectors, the distance between consecutive integers on the i-th axis line must be $1/\|\mathbf{v}_i\|$. Figure 1(c) shows a PCA plot of the Auto MPG data set with the additional axis lines. We used a standardized version of the data set to construct the plot (consecutive tick marks are separated by one standard deviation), but labeled the axes to reflect original non-standardized values.

2.4. Adaptable radial axes plots

Adaptable radial axes (ARA) plots [RSSL17] is a hybrid approach between SC and PCB. Similarly to SC, users can generate arbitrary linear mappings by selecting the axis vectors interactively. However, the embedded points are computed in order to optimize the approximations of high-dimensional data values (i.e., minimize estimation errors), similarly to biplots. Linear ARA plots (there are other types of nonlinear variants) are based on solving the following optimization problem:

$$\underset{\mathbf{P}\in\mathbb{R}^{N\times 2}}{\text{minimize}} \quad \|\mathbf{P}\mathbf{V}^{\mathsf{T}}-\mathbf{X}\|_{\mathrm{F}}^2. \tag{6}$$

In practice $\mathbf{X}$ should be replaced by its centered version $\mathbf{X}_c$, since this improves the approximations considerably. Thus, the data is typically standardized when applying the technique.

The objective function is identical to the one in (4), but in this case $\mathbf{V}$ and $\mathbf{X}$ are known, while $\mathbf{P}$ is the only variable. The solution is given by:

$$\mathbf{P}_* = \mathbf{X}[\mathbf{V}^{\dagger}]^{\mathsf{T}} = \mathbf{X}[\mathbf{V}^{\mathsf{T}}]^{\dagger}, \tag{7}$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse matrix. In the rest of the paper we will assume that the columns of $\mathbf{V}$ are linearly independent (i.e., $\mathbf{V}$ has rank 2, and $\mathbf{V}^{\mathsf{T}}\mathbf{V}$ is nonsingular), which occurs if at least two axis vectors point in different directions. In

that case (7) can be expressed as:

$$\mathbf{P}_* = \mathbf{X}[(\mathbf{V}^\mathsf{T}\mathbf{V})^{-1}\mathbf{V}^\mathsf{T}]^\mathsf{T} = \mathbf{X}\mathbf{V}(\mathbf{V}^\mathsf{T}\mathbf{V})^{-1}. \tag{8}$$

Note that if $\mathbf{V}$ is orthogonal an ARA plot is also an (orthographic) star coordinates plot. Figure 1(d) shows an example of an ARA plot with standardized data, for the same axis vectors used in Fig. 1(a). Similarly to PCB, the plot includes axis lines for approximating high-dimensional data values. It is also possible to incorporate axis lines in SC plots. However, the main advantage of ARA over SC is that its estimates of attribute values are usually considerably more accurate. Although ARA plots do not represent the data as well as PCB, they constitute a reasonable alternative when analysts must interact with or choose a desired collection of axis vectors (as in Figure 1(d) when searching for specific cars). Lastly, Scaled Radial Axes (SRA) [SSRMJ*18] is a variant of ARA that scales the axis vectors so that the label associated with the value 1 is located at the tip of the vector. The technique is useful for reducing overlaps, since in some cases it does not require depicting axis lines.

2.5. RadViz

A well-known radial technique related to SC is RadViz. It can be seen as a special case of SC if the data (which must be nonnegative) has been previously normalized so that the components of each sample add up to one [RSRDS16]. The motivation for RadViz stems from a physical spring system metaphor where the vectors $\mathbf{v}_i$ define anchor points for the springs rather than axis vectors or lines. Since RadViz is nonlinear and does not show vectors or line axes, we do not consider it to be a method based on radial axes in the context of this paper. However, in Sec. 3.2.4 we propose a practical application that incorporates vectors in RadViz visualizations.

3. Optimal axes for radial axes plots

In this section we present several approaches for improving SC and ARA plots regarding the ability to approximate data attributes (i.e., minimize estimation errors) by projecting embedded points onto labeled axes. The techniques generate or update alternative axis vectors and lines by solving diverse optimization problems.

3.1. Axis calibration (CAL)

In PCB the distances between consecutive integers on their i-th axis line must be $1/\|\mathbf{v}_i\|$. While this standard axis calibration is optimal for PCB, it is generally not for SC and ARA. Instead, assuming that the axes represent values on a linear scale, we can minimize the estimation errors for each variable independently by scaling and shifting its labels. Formally, the optimal scaling (α_i) and shift (β_i) for the i-th axis can be found by solving:

$$\underset{\alpha_i,\beta_i\in\mathbb{R}}{\text{minimize}} \quad \sum_{j=1}^{N} \left(\alpha_i(\mathbf{p}_j^\mathsf{T}\mathbf{v}_i) + \beta_i - x_{j,i}\right)^2, \tag{9}$$

where $\mathbf{p}_j$ is the j-th embedded point, and $x_{j,i}$ is the i-th attribute of the j-th data sample. Note that the estimate of $x_{j,i}$ is $\hat{x}_{j,i} = \alpha_i(\mathbf{p}_j^\mathsf{T}\mathbf{v}_i) + \beta_i$. The optimal solutions are (see the supplemental material):

$$\alpha_i^* = \frac{\sum_{j=1}^{N} x_{j,i}(\mathbf{p}_j^\mathsf{T}\mathbf{v}_i) - \bar{\mathbf{x}}_i \sum_{j=1}^{N} \mathbf{p}_j^\mathsf{T}\mathbf{v}_i}{\sum_{j=1}^{N} (\mathbf{p}_j^\mathsf{T}\mathbf{v}_i)^2 - \frac{1}{N}\left(\sum_{j=1}^{N} \mathbf{p}_j^\mathsf{T}\mathbf{v}_i\right)^2}, \tag{10}$$

and

$$\beta_i^* = \bar{\mathbf{x}}_i - \frac{\alpha_i^*}{N} \sum_{j=1}^{N} \mathbf{p}_j^\mathsf{T}\mathbf{v}_i, \tag{11}$$

where $\bar{\mathbf{x}}_i$ is the mean of the i-th data attribute. Note that with this basic approach the axis vectors and embedded points do not vary in the visualizations. Thus, the plot will remain consistent with the SC or ARA model described in (2) or (7). In the rest of the paper we will denote this approach as CAL.

Regarding the placement of the labels, the separation between integers on the i-th axis line is $1/(\alpha_i\|\mathbf{v}_i\|)$. Thus, the label for $\hat{x}_{j,i}$ should be placed at:

$$\frac{\mathbf{p}_j^\mathsf{T}\mathbf{v}_i}{\|\mathbf{v}_i\|} \cdot \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} = \frac{\hat{x}_{j,i} - \beta_i}{\alpha_i\|\mathbf{v}_i\|^2}\mathbf{v}_i. \tag{12}$$

Naturally, if $\hat{x}_{j,i}$ is a normalized value (typically obtained through some affine transformation), it would be necessary to undo the normalization in order to show the label for its original value.

Figure 2 illustrates how it is possible to enhance the estimates of high-dimensional data by labeling the axes adequately. The visualizations correspond to a SC plot of four variables of the Auto MPG data set that have been normalized to lie in $[0,1]$. In (a) we show the axis line associated with Displacement, whose horizontal axis vector has length $3/4$, and use the standard calibration procedure used in biplots and ARA. The labels associated with 0 and 1 appear at the origin, and at the point $\mathbf{v}/\|\mathbf{v}\|^2 = (4/3, 0)$, respectively, but we show the minimum (68) and maximum (455) unnormalized values of the variable. In this case the estimates, obtained by projecting embedded points orthogonally onto the axes, are poor. Since most of the plotted points lie to the left of the shaded region we would be underestimating the high-dimensional values (many estimates would be smaller than 68). In (b) we have scaled and shifted the axis optimally through the described calibration procedure, which improves the estimates considerably. Notice that the majority of the plotted points lie within the shaded region. In Sec. 3.2 we describe an alternative approach for improving the estimates.

In Tab. 1 we show the benefit of using the calibration approach regarding estimation accuracy, as well as several properties and relationships related to the radial axes methods. Since there is a lower bound on the approximation accuracy given by PCB, we show ratios of estimation errors for an entire data set ($\|\mathbf{P}\mathbf{V}^\mathsf{T} - \mathbf{X}\|_\mathrm{F}^2$) for a particular method, over the same quantity but for a PCB. We used five diverse data sets (Mice protein expression, SPECTF Heart - training, Ionosphere, Multiple Features - Fourier coefficients, Water Treatment Plant) available at [Lic13], where we normalized the variables to lie in $[0,1]$ and also discarded samples with missing values. The results are average ratios over 100 trials, where in each one we selected one of the data sets at random, and $n = 5$, 10 and

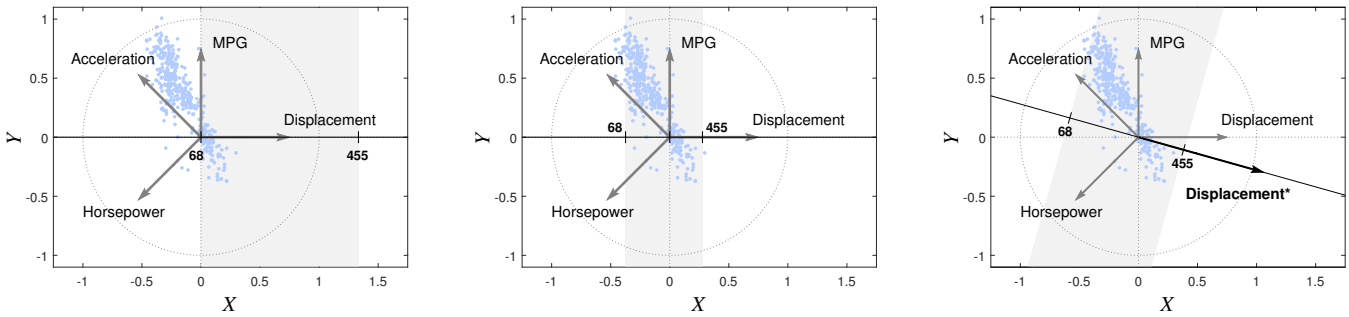

Figure 2: *Data approximation improvement when applying CAL and OPT. The graphics are SC plots of four variables of the Auto MPG data set, which have been normalized to lie in* $[0,1]$*. In (a) we have applied the standard calibration used in biplots and ARA to the Displacement axis, and labeled it with original (unnormalized) values. The minimum (68) is located at the origin, while the maximum (455) is further to the right. Since we obtain approximations by projecting embedded points orthogonally onto the axes, ideally these points should lie within the shaded region. Since most are outside of it the data estimates will be poor. In (b) we can enhance the estimation accuracy by applying the optimal calibration described in CAL, which shifts and scales the axes. Note that more plotted points lie inside the shaded region. In (c) we have applied OPT, which generates an axis with a different direction for which it is possible to enhance the approximations even further.*

	$n=5$				$n=10$				$n=15$			
	$\bar{\mathbf{x}} \neq \mathbf{0}$		$\bar{\mathbf{x}} = \mathbf{0}$		$\bar{\mathbf{x}} \neq \mathbf{0}$		$\bar{\mathbf{x}} = \mathbf{0}$		$\bar{\mathbf{x}} \neq \mathbf{0}$		$\bar{\mathbf{x}} = \mathbf{0}$	
	$\mathbf{V}$	$\mathbf{V}_\theta$	$\mathbf{V}$	$\mathbf{V}_\theta$	$\mathbf{V}$	$\mathbf{V}_\theta$	$\mathbf{V}$	$\mathbf{V}_\theta$	$\mathbf{V}$	$\mathbf{V}_\theta$	$\mathbf{V}$	$\mathbf{V}_\theta$
SC	395.2	39.07	47.87	4.389	879.2	83.71	160.2	14.22	2575	112.6	320.1	17.18
ARA or OSC	16.34		1.969		50.00		6.373		83.26		12.65	
SC + CAL	1.905				6.233				11.84			
(ARA or OSC) + CAL	1.723				5.505				11.31			
(SC, ARA, or OSC) + OPT	1.554				4.941				10.03			

Table 1: *Estimation error ratios. The values are averages of total estimation errors (*$\|\mathbf{P}\mathbf{V}^\top - \mathbf{X}\|_F^2$*) for a particular method and optimization strategy, divided by the minimal estimation error associated with a PCB. Also,* $\bar{\mathbf{x}} = 0$ *denotes centered data,* $\mathbf{V}$ *indicates a matrix with entries drawn from a standard normal distribution, and* $\mathbf{V}_\theta$ *is a scaled version of* $\mathbf{V}$ *according to (19). In this experiment we used five real data sets and computed the averages over 100 trials. In each one we selected n variables at random from one of the five data sets.*

15 of its variables also at random. We used two types of matrices of axis vectors: $\mathbf{V}$ and $\mathbf{V}_\theta$. The elements of $\mathbf{V}$ were drawn from a standard normal distribution. Alternatively, $\mathbf{V}_\theta$ (see Sec. (3.2.3)) is simply a scaled version of $\mathbf{V}$ that usually leads to more accurate estimates. We also ran experiments, denoted through $\bar{\mathbf{x}} = \mathbf{0}$, in which we centered the data in $[0,1]$.

In the first row we show the ratios when using SC and the standard calibration approach. It is apparent that centering the data is crucial, while scaling $\mathbf{V}$ can also affect the estimation accuracy considerably. For ARA and OSC scaling $\mathbf{V}$ does not affect the estimates, but centering the data is also critical. It is also worth noting that the estimates $(\hat{x}_{j,i})$ for ARA and OSC are identical, assuming the columns of $\mathbf{V}_\perp$ and $\mathbf{V}$ span the same subspace (see Prop. 1 in the supplemental material), which occurs when (3) holds.

Applying CAL enhances the estimates, which are better for ARA and OSC than for SC. This result should not be surprising, since the estimates for SC are considerably poorer than for ARA and OSC when applying the standard calibration used in biplots. In the next section we introduce an alternative approach that not only leads to better estimates, but these will be identical for the three analyzed radial methods.

3.2. Optimal axes for fixed embedded points (OPT)

In radial axes methods users specify axis vectors to indicate directions in which high-dimensional values should increase. However, since the location of an embedded point depends on the values and axis vectors of every variable, the orientations of the axis vectors will generally not reflect the best directions in which variable values are ordered (i.e., in which they increase). While in CAL the orientations of the axis vectors do not change, in this section we propose visualizing alternative axis vectors and lines that may be oriented differently, and which are optimal for approximating high-dimensional attribute values. Fig. 2(c) illustrates the idea, where the approach is able to find new axes for minimizing estimation errros even further.

3.2.1. Problem description

Given a data set $\mathbf{X}$ and a collection of embedded points $\mathbf{P}$, we model the approach through the following optimization problem:

$$\underset{\mathbf{v}_i \in \mathbb{R}^2,\, \gamma_i \in \mathbb{R}}{\text{minimize}} \quad \sum_{j=1}^{N} \left(\mathbf{p}_j^\top \mathbf{v}_i + \gamma_i - x_{j,i}\right)^2 , \tag{13}$$

where $\mathbf{v}_i$ is the optimal axis vector for the i-th data attribute. The second variable of the problem is the offset scalar γ_i that we introduce to shift the labels along its associated axis. This variable is necessary for obtaining optimal estimates for noncentered data. Note that ARA and biplots do not use these offsets and therefore require the data to be centered in order to produce optimal estimates. The solutions to the problem are (see the supplemental material):

$$\mathbf{v}_i^* = \mathbf{P}_c^\dagger \mathbf{x}_i, \tag{14}$$

and

$$\gamma_i^* = \bar{\mathbf{x}}_i - \frac{1}{N}\sum_{j=1}^{N} \mathbf{p}_j^\mathsf{T} \mathbf{v}_i^*, \tag{15}$$

where $\mathbf{P}_c$ is the $N \times 2$ matrix of centered plotted points, which we assume has rank 2. Considering all of the variables (14) can be expressed in matrix notation as:

$$\mathbf{V}_*^\mathsf{T} = \mathbf{P}_c^\dagger \mathbf{X}, \tag{16}$$

where $\mathbf{v}_i^*$ is the i-th row of $\mathbf{V}_*$. In the remainder of the paper we will denote this approach as OPT.

3.2.2. Comparison with CAL

In comparison with CAL, the approach provides more accurate approximations of high-dimensional data values since it can modify the orientation of the axis lines. Also, the lengths of the new axis vectors, together with the shift offsets, are also optimal regarding calibration. Note that OPT does not require scaling the dot products (through factors such as the α_i in CAL). Thus, the separation between integers on the i-th axis line will remain $1/\|\mathbf{v}_i^*\|$, as in ARA or biplots.

In Sec. 3.1 we saw that the approximation accuracy was identical for ARA and OSC after applying CAL (if $\mathscr{R}(\mathbf{V}) = \mathscr{R}(\mathbf{V}_\perp)$, where $\mathscr{R}$ denotes the range of a matrix), but poorer for SC. Although it would seem natural to expect similar results when applying OPT, the accuracy for SC is the same as that for ARA and OSC (see Prop. 2 and Cor. 1 in the supplemental material). Thus, in Tab. 1 we report the same value for the three methods after applying OPT. The main implication is that it is possible to mitigate the estimation accuracy limitation of SC by applying OPT, without the need to start with an ARA or OSC plot.

3.2.3. Guidelines for updating plots

Using the optimal axis vectors ($\mathbf{V}_*$) leads to more accurate estimates. Thus, the $\mathbf{v}_i^*$ reflect directions in which attribute values increase, or are ordered, better than the original vectors ($\mathbf{V}$). For example, they are generally more effective for determining the greatest and smallest data values. In addition, they can be visualized in order to guide users regarding how to update a plot. For instance, if there is a clear discrepancy between the directions of $\mathbf{v}_i$ and $\mathbf{v}_i^*$, users should seriously analyze the appropriateness of the orientation of $\mathbf{v}$, and consider replacing $\mathbf{v}$ by $\mathbf{v}^*$, or simply updating $\mathbf{v}$ towards $\mathbf{v}^*$. These updates generally (although not necessarily) improve the approximations of high-dimensional values.

Fig. 3 shows an example with the Olives data set [Lic13]. The graphic in (a) is a SC plot where the total squared estimation error is 965, while in (b) we have replaced the vector for Palmitic by its optimized counterpart, which is clearly different (the angle difference exceeds 90°). In this case the estimation error drops considerably to 221, which indicates that the initial choice of axis vector for Palmitic was poor regarding estimation accuracy. This is because Palmitic is negatively correlated with Oleic ($r = -0.84$), but positively correlated with Palmitoleic ($r = 0.84$).

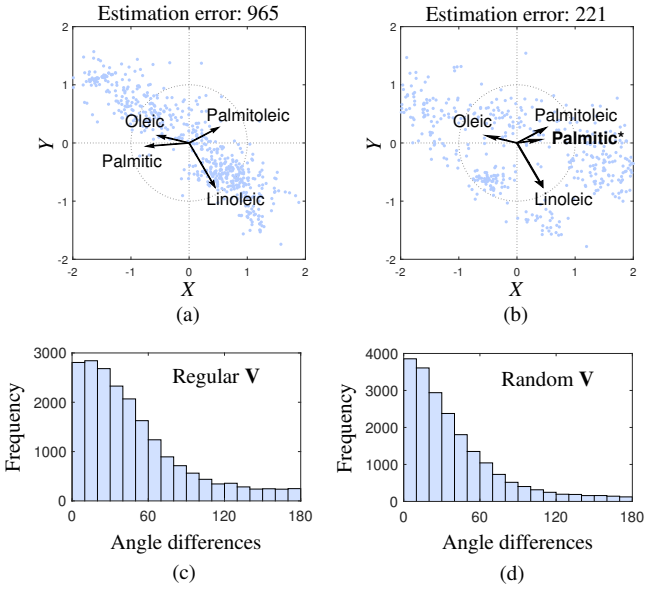

Figure 3: *Poor choices of axis vectors revealed by OPT, and distributions of absolute angle differences between axis vectors in* $\mathbf{V}$ *and* $\mathbf{V}_*$ *when applying OPT on the standardized Olives data set. An initial SC plot is shown in (a), where the total squared estimation error is 965 after applying CAL. In (b) we have replaced the Palmitic vector by its counterpart obtained through OPT, which is clearly different, and causes the error to drop to 272. In (c) and (d) we show distributions of absolute angle differences between vectors in* $\mathbf{V}$ *and* $\mathbf{V}_*$ *when applying OPT. In (c) we generated regular configurations of axis vectors for every ordering of the eight data variables, while in (d) we used random matrices whose elements were drawn from a standard normal distribution. In both cases there is an appreciable number of differences greater than* 90°.

For real data sets the orientation differences between $\mathbf{v}_i$ and $\mathbf{v}_i^*$ can be noteworthy in SC. In Fig. 3(c) and (d) we show distributions of angle differences between $\mathbf{v}_i$ by $\mathbf{v}_i^*$ in an experiment involving the eight (standardized) variables of the Olives data set. For the plot in (c) we generated the $7!/2$ different (excluding rotations and reflections) regular configurations of eight axis vectors ($\mathbf{v}_i = [\cos(i\pi/4), \sin(i\pi/4)]$), while in (d) we used $7!/2$ random matrices $\mathbf{V}$ whose elements were drawn from a standard normal distribution. Each histogram therefore reports $8!/2$ absolute angle differences between $\mathbf{v}_i$ by $\mathbf{v}_i^*$ when applying OPT. The percentage of differences greater than (90°) was 9.57% and 14.7% in (c) and (d), respectively. The histograms illustrate that for plots involving a moderate number of variables it is not uncommon to find at least one fairly large discrepancy in the orientations of the $\mathbf{v}_i$ and $\mathbf{v}_i^*$. Lastly, we obtained similar results with other real datasets.

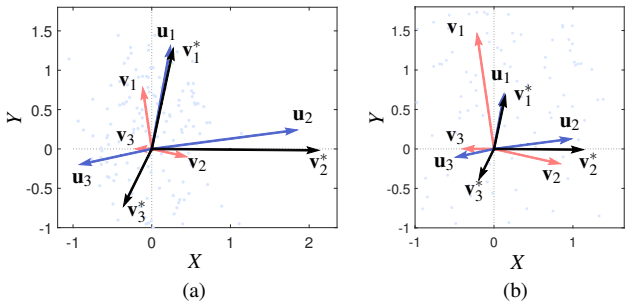

Figure 4: *Comparison of SC ($\mathbf{v}_i$), ARA ($\mathbf{u}_i$), and optimal ($\mathbf{v}_i^*$) axis vectors. Firstly, there is usually an inverse relation between the lengths of SC axis vectors and the lengths of $\mathbf{v}_i^*$. In this example the SC vectors in (b) are twice as long as in (a). Also, ARA axis vectors, which produce the same embedded points, are usually more similar to $\mathbf{v}_i^*$. Nevertheless, they can point towards notably different directions (in particular, $\mathbf{u}_3$ and $\mathbf{v}_3^*$ in the example).*

Since OPT only depends on $\mathbf{P}$ and $\mathbf{X}$, it is independent of the particular method used to obtain $\mathbf{P}$, whether it is SC, ARA, or any dimensionality reduction algorithm, which does not need to be based on radial axes necessarily (in Sec. 3.2.4 we present an example using RadViz). However, the new axis vectors would not reflect the underlying algorithm that generates the plotted points $\mathbf{P}$. Note that, since $\mathbf{P}$ and $\mathbf{X}$ are fixed, replacing $\mathbf{V}$ with $\mathbf{V}_*$ would lead to inconsistent SC or ARA models, i.e., (2) and (7) would not hold. Were we to generate a new set of embedded points through (7) for ARA, in accordance with $\mathbf{V}_*$, the approximation accuracy for the resulting (consistent) plot is guaranteed to improve. Specifically, applying (16) and (7) successively until convergence results in a PCB. This process is known as the alternating minimization algorithm (see [UHZB16]). We have also seen experimentally that applying (16) and (2) repeatedly also converges to a PCB. Furthermore, these iterative processes also converge to a PCB if we replace only one original axis vector (instead of the full matrix $\mathbf{V}$) by its optimal counterpart.

When deciding whether to replace some SC axis vector $\mathbf{v}_i$ with $\mathbf{v}_i^*$, it is convenient for them to have similar lengths. However, in SC the lengths of the optimal vectors are inversely related to the lengths of the original vectors. Note that if $\mathbf{v}_i$ is long the associated attribute values will be more spread out along the related axis. Consequently, $\|\mathbf{v}_i^*\|$ will be short, since units along an axis are separated by the inverse of the length of the corresponding axis vector. Figure 4 illustrates this inverse relation through two plots in which we have chosen a different scaling for the SC axis vectors ($\mathbf{v}_i$). We chose the ARA axis vectors, in this case denoted as $\mathbf{u}_i$, that would produce the same embedded points. Thus, the $n \times 2$ transformation matrix for ARA is $[\mathbf{V}^\dagger]^\top$, and the optimal axis vectors are the same for both methods. The example also shows that $\mathbf{v}_i^*$ is usually more similar to $\mathbf{u}_i$ than $\mathbf{v}_i$. The $\mathbf{v}_i^*$ can nevertheless have clearly different orientations, and therefore be useful for steering users towards plots that represent the data more faithfully (i.e., in which they can approximate data values with greater accuracy).

In order to mitigate the discrepancy between the lengths of $\mathbf{v}_i$ and $\mathbf{v}_i^*$, the matrix $\mathbf{V}$ can be scaled by some adequate factor θ. Specifically, we propose solving the following optimization problem:

$$\underset{\theta \in \mathbb{R}}{\text{minimize}} \quad \|\theta\mathbf{V} - \mathbf{V}_*^\theta\|_\mathrm{F}^2 \;, \tag{17}$$

where $\mathbf{V}_*^\theta$ represents the matrix of optimal axis vectors after having applied the scaling operation. For SC $\mathbf{V}_*^\theta = \mathbf{V}_*/\theta$, and the solution is (see the supplemental material):

$$\theta^* = \sqrt{\frac{\|\mathbf{V}_*\|_\mathrm{F}}{\|\mathbf{V}\|_\mathrm{F}}}. \tag{18}$$

Note that this operation (a basic zoom) preserves the ratio between the lengths of the axis vectors, and therefore users' intended relative importance of the variables. We will denote the scaled matrix as:

$$\mathbf{V}_\theta = \theta^*\mathbf{V}. \tag{19}$$

Lastly, the approach cannot be applied to ARA since in that case $\mathbf{V}_*^\theta = \theta\mathbf{V}_*$, and the solution to (17) would simply be $\theta = 0$.

The lengths of the optimal vectors can also provide information about the accuracy of the approximations, but analysts should be careful when interpreting them. When applying OPT on standardized data we usually observe an inverse relationship between $\|\mathbf{v}_i^*\|$ and the related estimation error (i.e, the objective function of (13)), which we will denote as ε_i. Figure 5(a) shows a histogram of Pearson correlation coefficients (r) between optimal axis vector lengths and associated estimation errors. Specifically, we generated 1000 plots of the five data sets used in the experiments reported in Tab. 1, but in this case we standardized the variables. For each plot we selected at random one of the data sets, 24 of its variables, and an initial matrix $\mathbf{V}$ with entries drawn from a standard normal distribution. Subsequently, we computed r with the pairs $(\|\mathbf{v}_i^*\|, \varepsilon_i)$, for $i = 1, \ldots, 24$. It is apparent that there is a clear negative correlation in the majority of the plots (although not in all of them). Lastly, we obtained similar results varying the number of variables and types of matrices $\mathbf{V}$ (regular, orthogonal, scaled, etc.).

Figure 5(b) shows the analogous correlations when normalizing the data to lie in the $[0,1]$ interval. In this case the distribution of correlations is approximately centered around 0. Thus, there is not a clear relationship between $\|\mathbf{v}_i^*\|$ and ε_i. Instead, there is usually a moderate positive correlation between the $\|\mathbf{v}_i^*\|$ and the variances of the data variables, as shown in Fig. 5(c).

The discrepancy between the distributions of correlations in Fig. 5(a) and (b) can be understood by examining the quadratic forms associated with $\|\mathbf{v}_i^*\|^2$ and ε_i. Firstly, note that:

$$\|\mathbf{v}_i^*\|^2 = \mathbf{x}_i^\top \underbrace{(\mathbf{P}_c^\dagger)^\top \mathbf{P}_c^\dagger}_{\succeq 0} \mathbf{x}_i = \mathbf{x}_{c,i}^\top \underbrace{(\mathbf{P}_c^\dagger)^\top \mathbf{P}_c^\dagger}_{\succeq 0} \mathbf{x}_{c,i},$$

where $\mathbf{x}_{c,i}$ is the centered version of the i-th data variable. Furthermore, ε_i can be written and decomposed as follows (see the supplemental material):

$$\varepsilon_i = \mathbf{x}_i^\top \underbrace{[\mathbf{C} - \mathbf{P}_c\mathbf{P}_c^\dagger]}_{\succeq 0} \mathbf{x}_i = N\sigma_i^2 + \mathbf{x}_{c,i}^\top \underbrace{[-\mathbf{P}_c\mathbf{P}_c^\dagger]}_{\preceq 0} \mathbf{x}_{c,i}.$$

Since $(\mathbf{P}_c^\dagger)^\top \mathbf{P}_c^\dagger$ is positive semidefinite $\|\mathbf{v}_i^*\|^2$ will generally tend to have greater values for larger absolute values in $\mathbf{x}_{c,i}$. This also occurs for ε_i, but notice that when the data is standardized all of

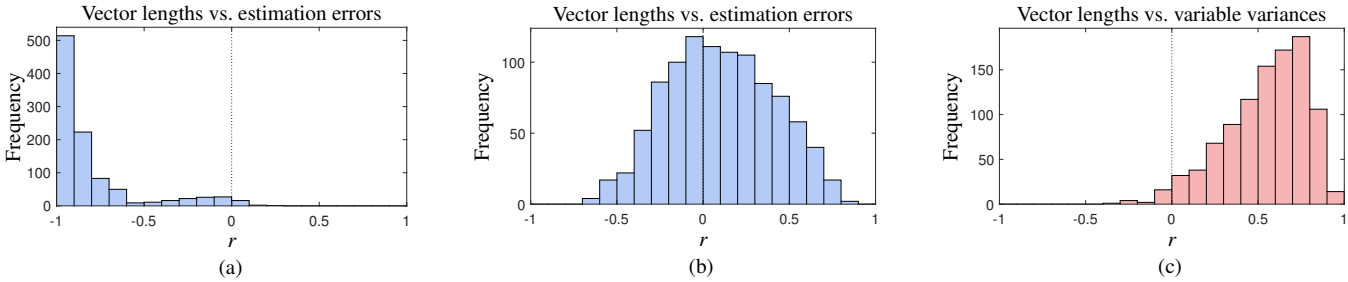

Figure 5: *Histograms of Pearson correlations (r) between the length of an optimal axis vector $\|\mathbf{v}_i^*\|$ and: (1) the estimation error, or (2) the variance, of the associated variable. In (a) we have used standardized data, which usually leads to strong negative correlations. In (b) and (c) we have used data normalized to lie in $[0,1]$. In this case, the $\|\mathbf{v}_i^*\|$ are mainly correlated with the variances of the variables.*

the variables have the same variance ($\sigma_i^2 = 1$). Thus, the term that characterizes the estimation error involves a negative semidefinite matrix $(-\mathbf{P}_c\mathbf{P}_c^\dagger)$, and ε_i will usually have smaller values for larger absolute values in $\mathbf{x}_{c,i}$. This would explain the clear negative correlations in Fig. 5(a). Similarly, for data normalized to lie in $[0,1]^n$ the variance term plays an important role in ε_i, which explains the positive correlations in Fig. 5(c).

3.2.4. Example: Application to RadViz

In this section we present an example that introduces optimal axis vectors in regular RadViz visualizations, where the anchors are placed evenly around a circle. We will use the axis vectors to assess the quality of variable orderings (i.e., the placement of variables along the circle), which is frequently addressed in the RadViz literature [CFMFM10, RDG12, RMK*14, BLH*14, Lon18, PT19, ABL*19].

In RadViz we can interpret that the anchor points associated with each variable pull the plotted data points towards them. Roughly speaking, the force of the pull is stronger for larger values of the associated attribute. Technically, RadViz differs from the radial methods analyzed in this paper in the sense that increasing the value of an attribute moves the plotted points towards the anchor point, and not in a specific direction. However, we will assume that on average the data values for a variable should increase in the direction from the origin to the anchor point (see the supplemental material).

We can evaluate whether the variable values increase in the direction of the anchors by incorporating optimal axis vectors in the visualizations. Figure 6 shows examples of these extended RadViz visualizations that use the Olives data set and a randomly selected ordering of its eight variables. Although the optimal vectors could be placed at the center of the RadViz plot, we position them with their origin at their associated anchor point. This avoids overlaps with the embedded points, which must appear within the convex hull of the set of anchor points.

The orientation of the vectors reveals the direction in which the values are optimally ordered, according to the criterion related to approximation accuracy in (13). In (a) the vector for the variable Arachidic is very well aligned with the direction of the anchor point. Therefore, the values of the variable, represented through the color coding, increase towards the bottom of the plot and are well-ordered in the visualization. Note that we are representing normalized values whose sum over all variables is equal to one. In contrast, in (b) the points are colored according to the values of the variable Palmitoleic. In this case not only are the values not ordered well, but the optimal axis vector points towards the interior of the RadViz visualization. Thus, the values for Palmitoleic generally tend to increase as they appear farther from its anchor. This clearly indicates that the anchor for Palmitoleic should be placed at another location around the circle. In (c) we have swapped the anchors for Palmitoleic and Palmitic, which improves the ordering of the plotted points with respect to Palmitoleic values. In addition, note that the distance from the embedded points to the anchors can be misleading when interpreting attribute values. Notice that the darker blue points that represent data samples with the largest Palmitoleic values are located near the center, while there are points with smaller values (to the left) that are closer to the variable's anchor. In this regard, the direction of the enhanced axis vector can help users avoid misinterpretations related to data values (we have included an enlarged red copy of the Palmitoleic axis vector, together with a perpendicular dashed line, simply for reference). Finally, the vector for Palmitic stands out for being very short. This does not mean that its estimates will be poor, as occurs in ARA. Instead, it simply indicates that the variance of the variable is likely to be small (see the supplemental material). In this example the variances of the rest of the variables are at least twice as large.

3.3. Optimal single vector updates

With OPT we obtained optimal axis vectors assuming the embedded points remain fixed. In this section we describe procedures for optimally modifying a single axis vector, or simply scaling it, but considering that the plotted points would change accordingly (i.e., applying the SC or ARA projection rules). These approaches guarantee decreasing the total estimation error, which does not occur necessarily in OPT if we replace a particular $\mathbf{v}_i$ by $\mathbf{v}_i^*$.

In order to simplify the proposed models and their solutions we will not consider axis offsets in the formulations, such as β_i and γ_i in (9) and (13), respectively. Notice that these factors do not affect the scaling constants α_i or optimal vectors $\mathbf{v}_i^*$ in CAL and OPT, and are only relevant for labeling the axes. Moreover, β_i and γ_i are simply introduced to account for data that is not centered (note that both are 0 if the data is centered). Similarly, the offsets will not

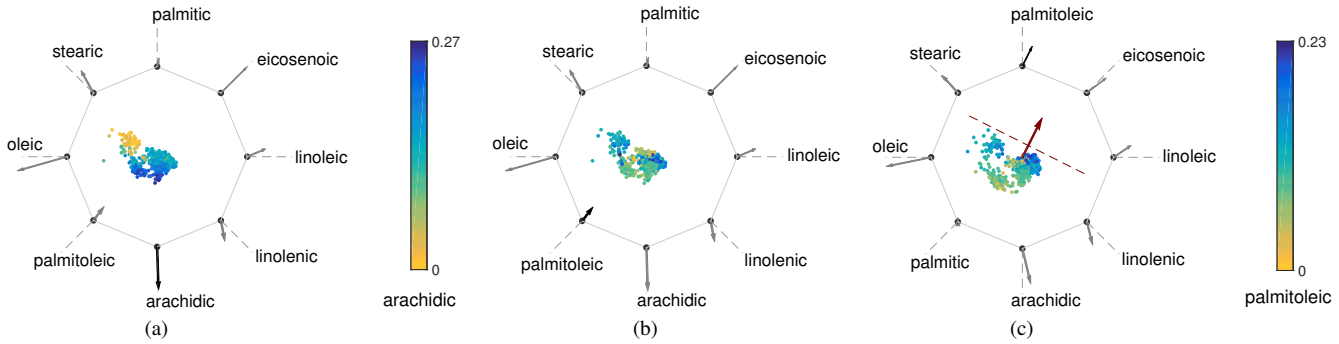

Figure 6: *Extended RadViz visualizations of the Olives data set. The plotted points are colored according to Arachidic values in (a), and the Palmitoleic attribute in (b) and (c), where darker blue dots represent large values (we consider normalized data where the sum of the attributes for each sample is one). In (a) the optimal axis vector for Arachidic is aligned with the direction of the anchor point, which indicates that its values appear well-ordered in the visualization. In (b) the enhanced axis vector for Palmitoleic points towards the interior of the RadViz visualization. Thus, its values generally decrease in the direction of the associated anchor. This suggests that the anchor for the variable should be located elsewhere in order to enhance the visualization. Also, the directions of the enhanced axis vectors are more reliable than the distances to the anchors for approximating data values. In (c) we have swapped the anchors for Palmitoleic and Palmitic. The darker blue points that have the larger values of Palmitoleic are not the closest ones to the corresponding anchor. These points lie to the right of the red arrow, which is simply an enlarged copy of the Palmitoleic enhanced axis vector. Lastly, we have drawn the perpendicular dashed line to the vector simply for reference.*

be necessary for the solutions in this section (see the supplemental material).

Given a radial axes plot for some data set $\mathbf{X}$, assume we would like to scale one axis vector by λ in order to generate a better plot (where $\mathbf{P}$ would change) regarding estimation accuracy. Without loss of generality, assume the vector to be scaled is the last one $\mathbf{v}_n$, and $\tilde{\mathbf{V}}$ the matrix of the first $n-1$ fixed axis vectors. In that case, we propose solving the following optimization problem:

$$\underset{\lambda \in \mathbb{R}}{\text{minimize}} \quad \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \lambda \mathbf{v}_n^{\mathsf{T}} \end{bmatrix} \mathbf{p}_j - \mathbf{x}_j \right\|^2 . \tag{20}$$

For SC, $\mathbf{p}_j = [\tilde{\mathbf{V}}^{\mathsf{T}},\ \lambda \mathbf{v}_n]\mathbf{x}_j$, and the objective function is a degree 4 polynomial $P(\lambda)$. Thus, the solution can be found reliably since it is a real root of the derivative of $P(\lambda)$ (see the supplemental material):

$$\begin{aligned} P'(\lambda) = {} & 4\lambda^3 \sum_{j=1}^{N} (\mathbf{v}_n^{\mathsf{T}} \mathbf{v}_n x_{j,n})^2 \ + \ 6\lambda^2 \sum_{j=1}^{N} \tilde{\mathbf{x}}_j^{\mathsf{T}} \tilde{\mathbf{V}} \mathbf{v}_n \mathbf{v}_n^{\mathsf{T}} \mathbf{v}_n x_{j,n} \\ & + 2\lambda \sum_{j=1}^{N} \left(\tilde{\mathbf{x}}_j^{\mathsf{T}} \tilde{\mathbf{V}} \mathbf{v}_n \mathbf{v}_n^{\mathsf{T}} \tilde{\mathbf{V}}^{\mathsf{T}} \tilde{\mathbf{x}}_j + (\mathbf{v}_n^{\mathsf{T}} \tilde{\mathbf{V}}^{\mathsf{T}} \tilde{\mathbf{V}} \mathbf{v}_n - 2\mathbf{v}_n^{\mathsf{T}} \mathbf{v}_n) x_{j,n}^2 \right) \\ & + 2 \sum_{j=1}^{N} \tilde{\mathbf{x}}_j^{\mathsf{T}} (\tilde{\mathbf{V}} \tilde{\mathbf{V}}^{\mathsf{T}} - 2\mathbf{I}) \tilde{\mathbf{V}} \mathbf{v}_n x_{j,n}, \end{aligned}$$

where $\tilde{\mathbf{x}}_j$ is the vector containing the first $n-1$ components of the j-th data sample (while $x_{j,n}$ is the corresponding n-th component).

For ARA the objective function is a more complicated quotient of polynomials, since $\mathbf{p}_j = [\tilde{\mathbf{V}}\ ;\ \lambda \mathbf{v}_n^{\mathsf{T}}]^{\dagger} \mathbf{x}_j$. Nevertheless, since it is a function of a single variable we can compute its solution through a basic direct search method or a more sophisticated approach (naturally, it is also possible to visualize the curve).

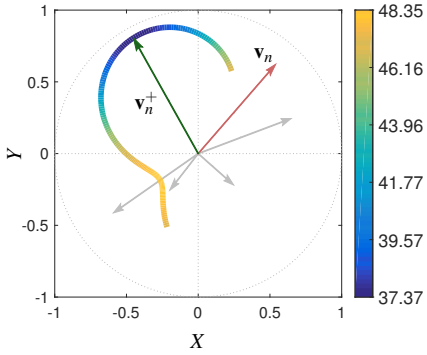

Figure 7: *Optimal length visualization for updated vectors. When choosing a different orientation for an axis vector ($\mathbf{v}_n$) it is possible to use the solution to (20) to obtain an optimal length for it. The colored curve included in the plot shows the optimal lengths for axis vectors orientated between 68° and 247°. It also shows the direction the axis vector should point towards to (e.g., a vector oriented at 30° ≡ 210° should end up pointing towards the 3rd quadrant). The colors indicate the estimation error that would result by choosing a vector with its endpoint on the curve. Thus, $\mathbf{v}_n^+$ is the optimal vector for replacing $\mathbf{v}_n$.*

We can use the solution to (20) not only to scale $\mathbf{v}_n$, but also to indicate the optimal length of any arbitrary axis vector that we could use instead of $\mathbf{v}_n$. Figure 7 illustrates this idea, where the colored curve indicates the optimal length of a vector that we could use to replace $\mathbf{v}_n$ in order to reduce the estimation error. We generated the curve by setting $\mathbf{v}_n = (\cos(\theta),\ \sin(\theta))$, for $\theta = 1°, \ldots, 180°$, solving (20), and plotting $\lambda \mathbf{v}_n$. Note that it is only necessary to use vectors for a half circle since λ can be negative. In the example,

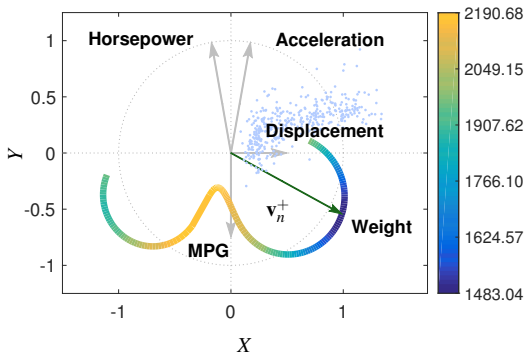

Figure 8: *Vector $\mathbf{v}_n^+$ when including the variable Weight in the plot in Fig. 1(a), which is more aligned with MPG and Displacement than with Horsepower and Acceleration.*

λ is negative for $\theta = 1°,\ldots,67°$. The color of the curve is the estimation error (i.e., the objective function). We have also depicted the optimal axis vector for reducing the estimation error, which we denote as $\mathbf{v}_n^+$.

It is also important to note that $\mathbf{v}_n$ could be the axis vector of a new variable that we wish to include in the plot. The colored curve would not only indicate the optimal coordinates of the axis vector ($\mathbf{v}_n^+$) for the new variable, it would also show the optimal vector length for other directions users may want to consider. Figure 8 shows $\mathbf{v}_n^+$ when including the variable Weight in the plot in Fig. 1(a). Although the plotted points move in the direction of the new vector, users will still be able to approximate data values reasonably, since the estimation error is minimal for that specific $\mathbf{v}_n^+$.

Lastly, a more elegant and precise way to compute $\mathbf{v}_n^+$ consists of solving:

$$\underset{\mathbf{v}_n \in \mathbb{R}^2}{\text{minimize}} \quad \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \mathbf{v}_n^\mathsf{T} \end{bmatrix} \mathbf{p}_j - \mathbf{x}_j \right\|^2 . \tag{21}$$

For SC we use a basic gradient descent method in order to find the solution. In our experiments the algorithm usually converges (very efficiently since the variable is a two-dimensional vector) with a fixed step size of $1/(100nN)$. The gradient of the objective function f_{SC} in (21) for SC is (see the supplemental material):

$$\nabla f_{\text{SC}} = 2\mathbf{C}\mathbf{v}_n + 2\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{a} + \left(2\mathbf{D} - 4\mathbf{I} + 4\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\right)\left(\mathbf{a} + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n\right), \tag{22}$$

where $\mathbf{x}_n$ is the n-th column of $\mathbf{X}$ (i.e., the vector containing the n-th attribute for every data sample), $\mathbf{D} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}$, $\mathbf{C} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\tilde{\mathbf{X}}\tilde{\mathbf{V}}$ (where $\tilde{\mathbf{X}}$ is the data matrix for the $n-1$ first variables), and $\mathbf{a} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\mathbf{x}_n$.

The gradient for ARA is far more complex. Thus, we have used a recursive search method to find the global minimum. Finally, in practice we run the search or gradient descent methods from four initial points belonging to each of the four quadrants on the plane.

4. Discussion

In this paper we have presented, compared, and analyzed in detail several approaches for updating and improving radial axes plots. The methods are based on optimization problems where the goal consists of finding alternative or new axis vectors and lines that better reflect the directions in which variable values are ordered and increase in the plots. The optimized quality metric is the discrepancy (in a least squares sense) between high-dimensional values and approximations or estimates of these, which are obtained by projecting embedded points onto labeled axes. Biplots or PCA also use this criterion. Besides helping users to obtain better approximations, the methods can be used to determine the orientation and length of the axis vectors, which is one of the main challenges users face when working with radial axes methods. In other words, they can help users construct or enhance visualizations, either by modifying axis vectors already present in the plots, or by suggesting the coordinates of new axis vectors (i.e., variables) to be included in the visualizations.

Regarding the optimal axis vectors described in OPT, showing them instead of the original axis vectors is generally not an issue. The original vectors convey information such as: (a) the displacement of an embedded point were we to increase the value of a variable by a unit, (b) a rough measure of the contribution of the variables to the plot, or (c) low-level intuition about the linear map (i.e., how a plot is constructed). However, this information is not useful for many analysis tasks. Moreover, the optimal vectors are better suited for searching for data with particular data values, analyzing the most common data values in clusters, inspecting outliers, etc.

The optimal axes can also be included in other plots. In particular, we showed how they can be used within RadViz to assess the quality of a variable ordering. This topic has been explored in many works in the literature. However, a comparison with these approaches, which use different criteria to define the quality of an arrangement, is well beyond the scope of the paper.

In practice, users must consider many factors when selecting the axis vectors, including the data analysis task, domain knowledge, the relative importance of each variable, the correlations between variables, etc. Our approaches are mainly concerned with enhancing the representation of high-dimensional data values. Thus, we do not consider tasks such as searching for cluster structure, separating classes, detecting outliers, etc. Nevertheless, since the methods suggest new coordinates for axis vectors, they also provide insight about the relevance of a variable (mainly through the length of the axis vector) or its correlation with the rest (through the orientation of the axis vector). We are planning on studying these relationships as future work.

Finally, while the paper proposes several algorithms based on optimization problems, it can also be considered as a theoretical work. The proofs and derivations provided in the supplemental material should be useful for visualization designers developing radial axes methods, or planning to incorporate axes into other visualization techniques.

Acknowledgements

This work was funded by the Spanish Ministry of Science and Innovation (grants RTI2018-098694-B-I00, PID2019-106623RB-C41, PID2019-107768RA-I00, PID2019-106623RB-C41) and by URJC and Community of Madrid (grant F661).

References

[ABL*19] ANGELINI M., BLASILLI G., LENTI S., PALLESCHI A., SANTUCCI G.: Towards enhancing radviz analysis and interpretation. In *2019 IEEE Visualization Conference (VIS)* (2019), pp. 226–230. doi:10.1109/VISUAL.2019.8933775. 8

[BLH*14] BINH H. T., LONG T. V., HOAI N. X., ANH N. D., TRUONG P. M.: Reordering dimensions for radial visualization of multidimensional data Û a genetic algorithms approach. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (2014), pp. 951–958. doi:10.1109/CEC.2014.6900619. 8

[CFMFM10] CARO L. D., FRIAS-MARTINEZ V., FRIAS-MARTINEZ E.: Analyzing the role of dimension arrangement for data visualization in radviz. In *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II* (2010), PAKDD'10, pp. 125–132. doi:10.1007/978-3-642-13672-6_13. 8

[DGRG12] DANIELS K. M., GRINSTEIN G. G., RUSSELL A., GLIDDEN M.: Properties of normalized radial visualizations. *Information Visualization 11*, 4 (2012), 273–300. doi:10.1177/1473871612439357. 2

[EMK*19] ESPADOTO M., MARTINS R. M., KERREN A., HIRATA N. S., TELEA A. C.: Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics* (2019). doi:10.1109/TVCG.2019.2944182. 1

[EY36] ECKART C., YOUNG G.: The approximation of one matrix by another of lower rank. *Psychometrika 1*, 3 (1936), 211–218. doi:10.1007/BF02288367. 3

[Gab71] GABRIEL K. R.: The biplot graphic display of matrices with application to principal component analysis. *Biometrika 58*, 3 (Dec 1971), 453–467. doi:10.1093/biomet/58.3.453. 1, 3

[GGLlR11] GOWER J., GARDNER-LUBBE S., LE ROUX N.: *Understanding Biplots*. John Wiley & Sons, 2011. 3

[GH95] GOWER J. C., HAND D. J.: *Biplots*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1995. 3

[GJH*01] GRINSTEIN G. G., JESSEE C. B., HOFFMAN P. E., O'NEIL P. J., GEE A. G.: High-dimensional visualization support for data mining gene expression data. In *DNA Arrays: Technologies and Experimental Strategies*, Grigorenko E. V., (Ed.). CRC Press LLC, Boca Raton, Florida, 2001, ch. 6, pp. 86–131. doi:10.1201/9781420038859.ch6. 2

[Gre10] GREENACRE M.: *Biplots in Practice*. BBVA Foundation, 2010. 3

[HGM*97] HOFFMAN P., GRINSTEIN G., MARX K., GROSSE I., STANLEY E.: DNA visual and analytic data mining. In *Proceedings of the 8th conference on Visualization '97* (Los Alamitos, CA, USA, 1997), VIS '97, IEEE Computer Society Press, pp. 437–441. doi:10.1109/VISUAL.1997.663916. 2

[Jol10] JOLLIFFE I. T.: *Principal component analysis*. Springer series in statistics. Springer-Verlag, 2010. 2

[Kan00] KANDOGAN E.: Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics* (2000), pp. 9–12. 1, 2

[Kan01] KANDOGAN E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2001), KDD'01, ACM, pp. 107–116. doi:10.1145/502512.502530. 2

[Lic13] LICHMAN M.: UCI machine learning repository, 2013. URL: archive.ics.uci.edu/ml. 2, 4, 6

[Lon18] LONG T. V.: Arcviz: An extended radial visualization for classes separation of high dimensional data. In *10th International Conference on Knowledge and Systems Engineering (KSE)* (2018), pp. 158–162. doi:10.1109/KSE.2018.8573428. 8

[LT13] LEHMANN D. J., THEISEL H.: Orthographic star coordinates. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (December 2013), 2615–2624. doi:10.1109/TVCG.2013.182. 1, 2

[LT16a] LEHMANN D. J., THEISEL H.: General projective maps for multidimensional data projection. *Computer Graphics Forum* (2016). doi:10.1111/cgf.12845. 2

[LT16b] LEHMANN D. J., THEISEL H.: Optimal sets of projections of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan 2016), 609–618. doi:10.1109/TVCG.2015.2467324. 2

[McL04] MCLACHLAN G. J.: *Discriminant analysis and statistical pattern recognition*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley-Interscience, 2004. doi:10.1002/0471725293. 2

[MH08] MAATEN L. V., HINTON G. E.: Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. 1

[MHSG18] MCINNES L., HEALY J., SAUL N., GROSSBERGER L.: UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software 3* (09 2018), 861. doi:10.21105/joss.00861. 1

[ML19] MOLCHANOV V., LINSEN L.: Shape-preserving star coordinates. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 449–458. doi:10.1109/TVCG.2018.2865118. 2

[PT19] PAGLIOSA L., TELEA A.: Radviz++: Improvements on radial-based visualizations. *Informatics 6*, 16 (04 2019). doi:10.3390/informatics6020016. 8

[RDG12] RUSSELL A., DANIELS K., GRINSTEIN G.: Voronoi diagram based dimensional anchor assessment for radial visualizations. In *Proceedings of the 2012 16th International Conference on Information Visualisation* (Washington, DC, USA, 2012), IV'12, IEEE Computer Society, pp. 229–233. doi:10.1109/IV.2012.46. 8

[RMK*14] RUSSELL A., MARCEAU R., KAMAYOU F., DANIELS K., GRINSTEIN G.: Clustered data separation via barycentric radial visualization. In *Proceedings of the 2014 International Conference on Modeling, Simulation and Visualization Methods (MSV)* (2014), pp. 101–Ű107. 8

[RSRDS16] RUBIO-SÁNCHEZ M., RAYA L., DÍAZ F., SANCHEZ A.: A comparative study between radviz and star coordinates. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (Jan 2016), 619–628. doi:10.1109/TVCG.2015.2467324. 4

[RSS14] RUBIO-SÁNCHEZ M., SANCHEZ A.: Axis calibration for improving data attribute estimation in star coordinates plots. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (Dec 2014), 2013–2022. doi:10.1109/TVCG.2014.2346258. 2

[RSSL17] RUBIO-SÁNCHEZ M., SANCHEZ A., LEHMANN D. J.: Adaptable radial axes plots for improved multivariate data visualization. *Computer Graphics Forum 36*, 3 (2017), 389–399. doi:10.1111/cgf.13196. 1, 3

[Sau20] SAUL L. K.: A tractable latent variable model for nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences 117*, 27 (2020), 15403–15408. doi:10.1073/pnas.1916012117. 1

[SGM08] SHARKO J., GRINSTEIN G., MARX K. A.: Vectorized radviz and its application to multiple cluster datasets. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov. 2008), 1444–1427. doi:10.1109/TVCG.2008.173. 2

[SSRMJ*18] SANCHEZ A., SOGUERO-RUIZ C., MORA-JIMÉNEZ I., RIVAS-FLORES F., LEHMANN D., RUBIO-SÁNCHEZ M.: Scaled radial axes for interactive visual feature selection: A case study for analyzing chronic conditions. *Expert Systems with Applications 100* (2018), 182–196.

doi:https://doi.org/10.1016/j.eswa.2018.01.054. 1, 4

[TLZM16] TANG J., LIU J., ZHANG M., MEI Q.: Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web - WWW'16* (2016), ACM Press, pp. 287Ű–297. doi:10.1145/2872427.2883041. 1

[UHZB16] UDELL M., HORN C., ZADEH R., BOYD S.: Generalized low rank models. *Foundations and Trends in Machine Learning 9*, 1 (June 2016), 1–118. doi:10.1561/2200000055. 7

[YMSJ05] YI J. S., MELTON R., STASKO J., JACKO J. A.: Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization 4*, 4 (2005), 239–256. doi:10.1057/palgrave.ivs.9500099. 14

Appendix A: Supplemental material

Optimal calibration scalings and shifts

Let f denote the objective function in (9). Firstly, setting the partial derivative of f with respect to β_i equal to 0 we have:

$$\frac{\partial f}{\partial \beta_i} = 2\sum_{j=1}^{N}\left(\alpha_i \mathbf{p}_j^\top \mathbf{v}_i + \beta_i - x_{j,i}\right) = 0.$$

Solving for β_i yields (11):

$$\beta_i = \bar{\mathbf{x}}_i - \frac{\alpha_i}{N}\sum_{j=1}^{N}\mathbf{p}_j^\top \mathbf{v}_i.$$

Secondly, setting the partial derivative of f with respect to α_i equal to 0 we have:

$$\frac{\partial f}{\partial \alpha_i} = 2\sum_{j=1}^{N}\left(\alpha_i \mathbf{p}_j^\top \mathbf{v}_i + \beta_i - x_{j,i}\right)\mathbf{p}_j^\top \mathbf{v}_i = 0.$$

Simplifying and incorporating the expression for β_i we have:

$$\alpha_i \sum_{j=1}^{N}(\mathbf{p}_j^\top \mathbf{v}_i)^2 + \left(\bar{\mathbf{x}}_i - \frac{\alpha_i}{N}\sum_{j=1}^{N}\mathbf{p}_j^\top \mathbf{v}_i\right)\left(\sum_{j=1}^{N}\mathbf{p}_j^\top \mathbf{v}_i\right) - \sum_{j=1}^{N} x_{j,i}(\mathbf{p}_j^\top \mathbf{v}_i) = 0.$$

Finally, solving for α_i yields (10):

$$\alpha_i = \frac{\sum_{j=1}^{N} x_{j,i}(\mathbf{p}_j^\top \mathbf{v}_i) - \bar{\mathbf{x}}_i \sum_{j=1}^{N}\mathbf{p}_j^\top \mathbf{v}_i}{\sum_{j=1}^{N}(\mathbf{p}_j^\top \mathbf{v}_i)^2 - \frac{1}{N}\left(\sum_{j=1}^{N}\mathbf{p}_j^\top \mathbf{v}_i\right)^2}.$$

Identical approximation accuracy for OSC and ARA when applying CAL

Proposition 1 Consider applying CAL on an OSC plot with matrix $\mathbf{V}_\perp$, and an ARA plot with matrix $\mathbf{V}$. If $\mathbf{V}_\perp$ and $\mathbf{V}$ span the same subspace (i.e., if $\mathscr{R}(\mathbf{V}) = \mathscr{R}(\mathbf{V}_\perp)$), the estimates $\hat{x}_{j,i} = \alpha_i(\mathbf{p}_j^\top \mathbf{v}_i) + \beta_i$ are identical in both plots, where $\mathbf{v}_i$ denotes the i-th axis vector in either method.

Proof The proposition holds since the dot products $\mathbf{p}_j^\top \mathbf{v}_i$ are the same in both methods, which also implies that CAL will find identical values of α_i and β_i for a given data set $\mathbf{X}$. The values $\mathbf{p}_j^\top \mathbf{v}_i$ are the entries of the vector of dot products $\mathbf{V}\mathbf{p}$, which is the orthogonal projection of the data sample $\mathbf{x}_j$ onto $\mathscr{R}(\mathbf{V})$, and therefore identical in both methods.
For example, in ARA we have:

$$\mathbf{V}\mathbf{p} = \mathbf{V}(\mathbf{V}^\top \mathbf{V})^{-1}\mathbf{V}^\top \mathbf{x},$$

while in OSC:

$$\mathbf{V}_\perp \mathbf{p} = \mathbf{V}_\perp \mathbf{V}_\perp^\top \mathbf{x} = \mathbf{V}_\perp(\mathbf{V}_\perp^\top \mathbf{V}_\perp)^{-1}\mathbf{V}_\perp^\top \mathbf{x}.$$

Recall that $\mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{x}$ is the orthogonal projection of $\mathbf{x}$ onto $\mathscr{R}(\mathbf{A})$. Since we have assumed that $\mathscr{R}(\mathbf{V}) = \mathscr{R}(\mathbf{V}_\perp)$ it follows that $\mathbf{V}\mathbf{p} = \mathbf{V}_\perp \mathbf{p}$. □

Solutions for optimal axes

In this section we show that the solutions to (13) are given by (14) and (15).

Firstly, the objective function in (13) can be rewritten as:

$$\begin{aligned} f_i(\mathbf{v}_i, \gamma_i) &= \sum_{j=1}^{N}\left(\mathbf{p}_j^\top \mathbf{v}_i + \gamma_i - x_{j,i}\right)^2 \\ &= \|\mathbf{P}\mathbf{v}_i + \gamma_i \mathbf{1} - \mathbf{x}_i\|^2 \\ &= (\mathbf{P}\mathbf{v}_i + \gamma_i \mathbf{1} - \mathbf{x}_i)^\top (\mathbf{P}\mathbf{v}_i + \gamma_i \mathbf{1} - \mathbf{x}_i) \\ &= \mathbf{v}_i^\top \mathbf{P}^\top \mathbf{P}\mathbf{v}_i - 2\mathbf{v}_i^\top \mathbf{P}^\top \mathbf{x}_i + 2\gamma_i \mathbf{v}_i^\top \mathbf{P}^\top \mathbf{1} \\ &\quad + N\gamma_i^2 - 2\gamma_i \mathbf{x}_i^\top \mathbf{1} + \mathbf{x}_i^\top \mathbf{x}_i, \end{aligned}$$

where $\mathbf{P}$ is the $N \times 2$ matrix of plotted points (not necessarily centered), $\mathbf{1}$ is a vector of N ones, and $\mathbf{x}_i$ is the N-dimensional vector of attribute values for the i-th data variable.

The partial derivatives with respect to γ_i and $\mathbf{v}_i$ are:

$$\frac{\partial f_i}{\partial \gamma_i} = -2\mathbf{v}_i^\top \mathbf{P}^\top \mathbf{1} + 2N\gamma_i + 2\mathbf{x}_i^\top \mathbf{1}, \tag{23}$$

and

$$\frac{\partial f_i}{\partial \mathbf{v}_i} = 2\mathbf{P}^\top \mathbf{P}\mathbf{v}_i - 2\mathbf{P}^\top \mathbf{x}_i - 2\gamma_i \mathbf{P}^\top \mathbf{1}. \tag{24}$$

Setting (23) to 0 yields:

$$\begin{aligned} \gamma_i &= \frac{1}{N}(\mathbf{x}_i^\top - \mathbf{v}_i^\top \mathbf{P}^\top)\mathbf{1} = \frac{1}{N}\mathbf{1}^\top(\mathbf{x}_i - \mathbf{P}\mathbf{v}_i) \\ &= \frac{1}{N}\sum_{j=1}^{N}\left(x_{j,i} - \mathbf{p}_j^\top \mathbf{v}_i\right) = \bar{\mathbf{x}}_i - \frac{1}{N}\sum_{j=1}^{N}\mathbf{p}_j^\top \mathbf{v}_i. \end{aligned} \tag{25}$$

Substituting the expression for γ_i in (24) and setting the partial derivative to 0 yields:

$$\mathbf{P}^\top \mathbf{P}\mathbf{v}_i - \mathbf{P}^\top \mathbf{x}_i - \frac{1}{N}\mathbf{1}^\top(\mathbf{P}\mathbf{v}_i - \mathbf{x}_i)\mathbf{P}^\top \mathbf{1} = \mathbf{0}.$$

Since $\mathbf{1}^\top \mathbf{P}\mathbf{v}_i$ and $\mathbf{1}^\top \mathbf{x}_i$ are scalars we can write the equation as:

$$\mathbf{P}^\top \mathbf{P}\mathbf{v}_i - \mathbf{P}^\top \mathbf{x}_i - \frac{1}{N}\mathbf{P}^\top \mathbf{1}\mathbf{1}^\top \mathbf{P}\mathbf{v}_i + \frac{1}{N}\mathbf{P}^\top \mathbf{1}\mathbf{1}^\top \mathbf{x}_i = \mathbf{0},$$

$$\mathbf{P}^\top\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\mathbf{P}\mathbf{v}_i = \mathbf{P}^\top\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\mathbf{x}_i,$$

where $\mathbf{I}$ is the $N \times N$ identity matrix. Additionally, $\mathbf{I} - (1/N)\mathbf{1}\mathbf{1}^\top$ is the well-known "centering" matrix, which is symmetric and idempotent. Thus, we can rewrite the previous equation as:

$$\mathbf{P}_c^\top \mathbf{P}_c \mathbf{v}_i = \mathbf{P}_c^\top \mathbf{x}_i,$$

where $\mathbf{P}_c = (\mathbf{I} - (1/N)\mathbf{1}\mathbf{1}^\top)\mathbf{P}$ is the centered version of $\mathbf{P}$ (i.e., its column sums are 0). Also, note that the data samples can also be centered, in which case the embedded points of any linear transformation will also be centered. Finally, assuming $\mathbf{P}_c$ has rank 2 we have:

$$\mathbf{v}_i = (\mathbf{P}_c^\top \mathbf{P}_c)^{-1}\mathbf{P}_c^\top \mathbf{x}_i = \mathbf{P}_c^\dagger \mathbf{x}_i.$$

Identical approximation accuracy for SC, OSC, and ARA when applying OPT

Proposition 2 Let $\mathbf{X}$ represent an $N \times n$ data matrix, $\mathbf{V}$ an $n \times 2$ matrix of full column rank, and $\mathbf{M}$ a 2×2 invertible matrix. In addition, let $\mathbf{P}$ denote the $N \times 2$ matrix that is the result of mapping the N data samples of $\mathbf{X}$ linearly onto a plane through the matrix $\mathbf{VM}$. In other words:

$$\mathbf{P} = \mathbf{XVM}. \tag{26}$$

Furthermore, let $\mathbf{V}_* = \mathbf{P}^\dagger \mathbf{X}$ denote a matrix of enhanced axis vectors as defined in (16). Finally, let $\mathbf{P}\mathbf{V}_*^\mathsf{T}$ represent approximations of data samples in $\mathbf{X}$ by projecting the N embedded points in $\mathbf{P}$ orthogonally onto the enhanced axes, as defined in biplots or ARA plots (i.e., the approximations are the dot products between the embedded points and the enhanced axis vectors). In that case, the approximations $\mathbf{P}\mathbf{V}_*^\mathsf{T}$ do not depend on $\mathbf{M}$.

Proof Firstly, $\mathbf{V}_* = \mathbf{P}^\dagger \mathbf{X} = [\mathbf{XVM}]^\dagger \mathbf{X}$. Thus, we can express the approximations as:

$$\mathbf{P}\mathbf{V}_*^\mathsf{T} = \mathbf{XVM}(\mathbf{M}^\mathsf{T}\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{XVM})^{-1}\mathbf{M}^\mathsf{T}\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}.$$

Since $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ for $m \times m$ invertible square matrices $\mathbf{A}$ and $\mathbf{B}$, we can rewrite the approximations as:

$$\begin{aligned}\mathbf{P}\mathbf{V}_*^\mathsf{T} &= \mathbf{XVMM}^{-1}(\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{XV})^{-1}(\mathbf{M}^\mathsf{T})^{-1}\mathbf{M}^\mathsf{T}\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X} \\ &= \mathbf{XV}(\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{XV})^{-1}\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T} = (\mathbf{XV})(\mathbf{XV})^\dagger\mathbf{X}.\end{aligned}$$

Thus, they do not depend on matrix $\mathbf{M}$. □

Corollary 1 Let $\mathbf{X}$ represent an $N \times n$ data matrix, $\mathbf{V}$ an $n \times 2$ matrix of full column rank, and $\mathbf{V}_\perp$ and orthogonal matrix with the same range as $\mathbf{V}$. In addition, consider mapping the data samples in $\mathbf{X}$ onto a plane with SC, ARA and OSC, through (2), (7), and (3), respectively. The approximations of $\mathbf{X}$ resulting from projecting the embedded points orthogonally onto enhanced labeled axes, as they are defined in biplots or ARA plots, and which are obtained through (16), are identical for the three methods.

Proof The mappings for SC, ARA and OSC all have the form in (26). In particular, for SC $\mathbf{M} = \mathbf{I}$, for ARA $\mathbf{M} = (\mathbf{V}^\mathsf{T}\mathbf{V})^{-1}$, and for OSC $\mathbf{M} = \mathbf{B}$, where $\mathbf{V}_\perp = \mathbf{VB}$. Therefore, due to Prop. 2, the approximations when using the enhanced axes are identical for the three methods. □

Solution for θ^*

We now show that the solution to (17) is given by (18). Firstly, recall that $\mathbf{P} = \mathbf{XV}$ is the set of embedded points prior to performing the scaling by θ, and $\mathbf{P}_c = \mathbf{CXV}$ contains the corresponding centered points, where $\mathbf{C} = \mathbf{I} - (1/N)\mathbf{1}\mathbf{1}^\mathsf{T}$ is the symmetric and idempotent centering matrix. Similarly, we denote the set of embedded points after performing the scaling as $\mathbf{P}^\theta = \theta\mathbf{XV}$, while $\mathbf{P}_c^\theta = \theta\mathbf{CXV}$ is its centered version.

The objective function of the optimization problem can be written as:

$$\begin{aligned}f(\theta) &= \|\theta\mathbf{V} - \mathbf{V}_*^\theta\|_\mathrm{F}^2 = \|\theta\mathbf{V}^\mathsf{T} - (\mathbf{P}_c^\theta)^\dagger\mathbf{X}\|_\mathrm{F}^2 \\ &= \|\theta\mathbf{V}^\mathsf{T} - (\theta\mathbf{CXV})^\dagger\mathbf{X}\|_\mathrm{F}^2 \\ &= \|\theta\mathbf{V}^\mathsf{T} - (\theta^2\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{C}^2\mathbf{XV})^{-1}\theta\mathbf{V}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{CX}\|_\mathrm{F}^2 \\ &= \|\theta\mathbf{V}^\mathsf{T} - \frac{1}{\theta}\mathbf{P}_c^\dagger\mathbf{X}\|_\mathrm{F}^2 = \|\theta\mathbf{V}^\mathsf{T} - \frac{1}{\theta}\mathbf{V}_*^\mathsf{T}\|_\mathrm{F}^2 \\ &= \mathrm{tr}\left[\left(\theta\mathbf{V}^\mathsf{T} - \frac{1}{\theta}\mathbf{V}_*^\mathsf{T}\right)^\mathsf{T}\left(\theta\mathbf{V}^\mathsf{T} - \frac{1}{\theta}\mathbf{V}_*^\mathsf{T}\right)\right] \\ &= \theta^2\mathrm{tr}(\mathbf{V}^\mathsf{T}\mathbf{V}) - 2\mathrm{tr}(\mathbf{V}_*^\mathsf{T}\mathbf{V}) + \frac{1}{\theta^2}\mathrm{tr}\left(\mathbf{V}_*^\mathsf{T}\mathbf{V}_*\right) \\ &= \theta^2\|\mathbf{V}\|_\mathrm{F}^2 - 2\mathrm{tr}(\mathbf{V}_*^\mathsf{T}\mathbf{V}) + \frac{1}{\theta^2}\|\mathbf{V}_*\|_\mathrm{F}^2,\end{aligned}$$

where tr denotes the trace of a matrix. Setting its derivative equal to zero yields:

$$f'(\theta) = 2\theta\|\mathbf{V}\|_\mathrm{F}^2 - \frac{2}{\theta^3}\|\mathbf{V}_*\|_\mathrm{F}^2 = 0.$$

Finally, solving for θ we have:

$$\theta^* = \sqrt[4]{\frac{\|\mathbf{V}_*\|_\mathrm{F}^2}{\|\mathbf{V}\|_\mathrm{F}^2}} = \sqrt{\frac{\|\mathbf{V}_*\|_\mathrm{F}}{\|\mathbf{V}\|_\mathrm{F}}}.$$

Relationship between accuracy and axis vector length in ARA

There is a direct relationship between approximation accuracy and axis vector length in ARA plots. Since integers on the i-th line axis are located at multiples of $1/\|\mathbf{v}_i\|$, they appear closer to each other for larger axis vectors. This implies that a variation in $\mathbf{p}$ in the direction of a large axis vector will cause a larger approximation error for the variable. Thus, the method will primarily focus on minimizing the approximation errors for variables with larger axis vectors. In Fig. 9 we illustrate this effect by comparing an initial ARA plot to another in which we have enlarged one axis vector. The example is based on the standardized Breakfast cereal data set used in [YMSJ05], but have labeled the axes with original data values. In this case, when an axis vector associated with Calories is stretched the plotted points appear more compacted in the direction of the axis (in SC the effect would be the opposite). Furthermore, the corresponding approximations are more accurate. Thus, the plotted points appear better ordered in the direction of the axis, as can be seen through the color coding of the dots, which represents caloric content.

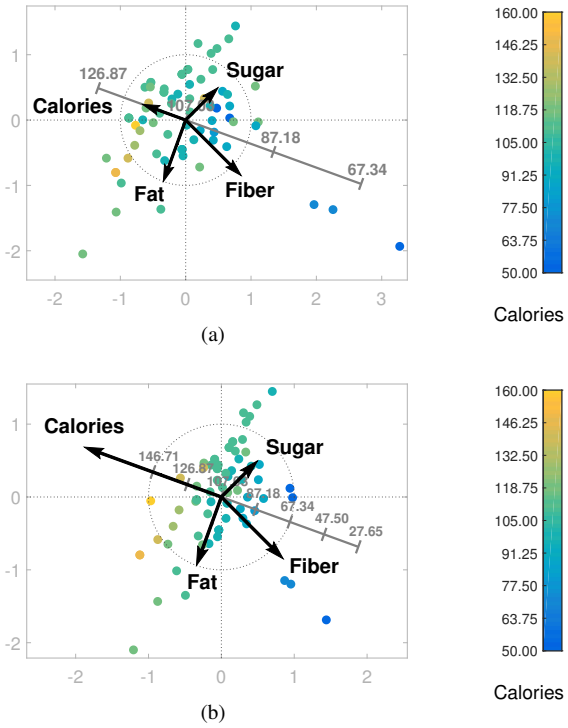

Figure 9: *Direct relationship between accuracy and axis vector length in ARA plots. In this example the only difference between the ARA plots in (a) and (b) is that the axis vector associated with the variable Calories is longer in the latter. This compresses the plotted points along the direction of the axis in (b), and improves the approximation accuracy for Calories. In particular, observe (through the color coding) that the points appear better ordered with respect to caloric content along the direction of the axis.*

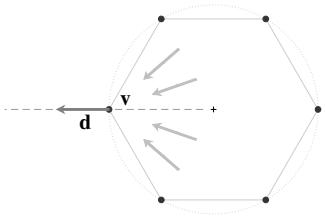

Figure 10: *In RadViz increasing the value of a data variable causes the plotted points to move towards the anchor point* $\mathbf{v}$ *associated with the variable. On average we assume that we should expect to find greater values for the variable in the direction* ($\mathbf{d}$) *from the origin towards* $\mathbf{v}$.

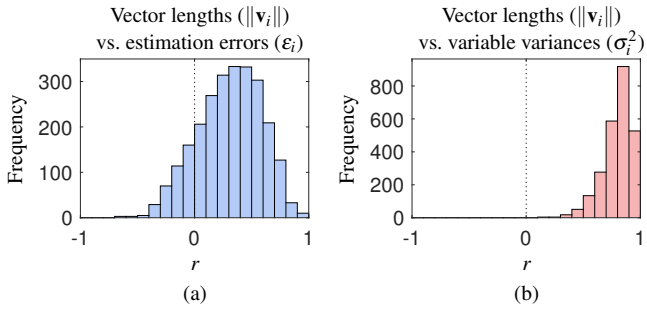

Figure 11: *Histograms of Pearson correlations (r) between the length of an optimal axis vector and the estimation error, and the variance, of the associated variable, for the data used in the RadViz example in Sec. 3.2.4. The variables were first normalized to lie in* $[0,1]$, *and afterwards for each sample we divided each attribute by the sum of all of the attributes. Since each variable has a different variance,* $\|\mathbf{v}_i\|$ *is usually positively correlated with* σ_i^2, *but not necessarily with* ε_i.

Decomposition of the estimation errors ε_i

The objective function in (13), denoted here as ε_i can be rewritten as follows:

$$\begin{aligned}
\varepsilon_i &= \sum_{j=1}^{N} \left(\mathbf{p}_j^\top \mathbf{v}_i^* + \gamma_i^* - x_{j,i}\right)^2 = \|\mathbf{P}\mathbf{v}_i^* + \mathbf{1}\gamma_i^* - \mathbf{x}_i\|^2 \\
&= \left\| \mathbf{P}\mathbf{P}_c^\dagger \mathbf{x}_i + \mathbf{1}\left(\frac{1}{N}\mathbf{1}^\top \mathbf{x}_i - \frac{1}{N}\mathbf{1}^\top \mathbf{P}\mathbf{P}_c^\dagger \mathbf{x}_i\right) - \mathbf{x}_i \right\|^2 \\
&= \left\| (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top)\mathbf{P}\mathbf{P}_c^\dagger \mathbf{x}_i - (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top)\mathbf{x}_i \right\|^2 \\
&= \|(\mathbf{P}_c\mathbf{P}_c^\dagger - \mathbf{C})\mathbf{x}_i\|^2 \\
&= \mathbf{x}_i^\top (\mathbf{P}_c\mathbf{P}_c^\dagger - \mathbf{C})^\top (\mathbf{P}_c\mathbf{P}_c^\dagger - \mathbf{C})\mathbf{x}_i \\
&= \mathbf{x}_i^\top (\mathbf{C} - \mathbf{P}_c\mathbf{P}_c^\dagger)\mathbf{x}_i = \mathbf{x}_i^\top \mathbf{C}\mathbf{x}_i - \mathbf{x}_i^\top \mathbf{P}_c\mathbf{P}_c^\dagger \mathbf{x}_i \\
&= N\sigma_i^2 - \mathbf{x}_i^\top \mathbf{C}\mathbf{P}_c\mathbf{P}_c^\dagger \mathbf{C}\mathbf{x}_i \\
&= N\sigma_i^2 - \mathbf{x}_{c,i}^\top \mathbf{P}_c\mathbf{P}_c^\dagger \mathbf{x}_{c,i}.
\end{aligned}$$

Effect of increasing an attribute in RadViz

Figure 10 shows the effect of increasing an attribute value in RadViz. The plotted point moves towards the anchor associated with the corresponding variable. We assume that on average the data values for a variable should increase in the direction ($\mathbf{d}$) from the origin to the anchor point $\mathbf{v}$.

Correlations related to $\|\mathbf{v}_i\|$ for the RadViz example

In the RadViz example (see Sec. 3.2.4) the length of the optimal vectors predominantly reflects the variance of the variables. In Fig. 11 we show distributions of correlations between $\|\mathbf{v}_i\|$, and ε_i and σ_i^2, for the $7!/2$ different orderings of the eight variables (discarding rotations and reflections). The vector lengths usually have a strong positive correlation with the variable variances. In this example the variance for Palmitic is 0.0009, which is at least twice as small as the rest of the variances, which explains the short length of the Palmitic axis vector in Fig. 6.

Discarding offset shifts in the objective functions

The objective functions of the optimization problems considered in this paper have the following form:

$$f = \|\mathbf{P}\mathbf{V}^\mathsf{T} - \mathbf{X} + \mathbf{1}\boldsymbol{\delta}^\mathsf{T}\|_\mathrm{F}^2, \tag{27}$$

where the variables related to the axis vectors appear in $\mathbf{V}$ and $\mathbf{P}$. Also, assume $\mathbf{V}$, $\mathbf{P}$, and $\boldsymbol{\delta}$ are or contain optimal solutions. In that case $\boldsymbol{\delta}$ is (see the derivation of (25)):

$$\boldsymbol{\delta} = \bar{\mathbf{x}} - \mathbf{V}\bar{\mathbf{p}} = -(\mathbf{V}\mathbf{W} - \mathbf{I})\bar{\mathbf{x}},$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{p}}$ are the mean of the data and plotted points, respectively. Additionally, $\mathbf{W} = \mathbf{V}^\mathsf{T}$ for SC, and $\mathbf{W} = \mathbf{V}^\dagger$ for ARA. Substituting in (27) we can rewrite the objective function as:

$$\begin{aligned} f &= \|\mathbf{V}\mathbf{P}^\mathsf{T} - \mathbf{X}^\mathsf{T} + \boldsymbol{\delta}\mathbf{1}^\mathsf{T}\|_\mathrm{F}^2 \\ &= \|\mathbf{V}\mathbf{W}\mathbf{X}^\mathsf{T} - \mathbf{X}^\mathsf{T} - (\mathbf{V}\mathbf{W} - \mathbf{I})\bar{\mathbf{x}}\mathbf{1}^\mathsf{T}\|_\mathrm{F}^2 \\ &= \|(\mathbf{V}\mathbf{W} - \mathbf{I})\mathbf{X}^\mathsf{T} - (\mathbf{V}\mathbf{W} - \mathbf{I})\bar{\mathbf{x}}\mathbf{1}^\mathsf{T}\|_\mathrm{F}^2 \\ &= \|(\mathbf{V}\mathbf{W} - \mathbf{I})(\mathbf{X}^\mathsf{T} - \bar{\mathbf{x}}\mathbf{1}^\mathsf{T})\|_\mathrm{F}^2. \end{aligned}$$

If we apply a translation $\mathbf{s}$ to the data, the new data matrix would become $\mathbf{X} + \mathbf{1}\mathbf{s}^\mathsf{T}$, while the new mean would be $\bar{\mathbf{x}} - \mathbf{s}$. In that case, f would not change:

$$\begin{aligned} f &= \|(\mathbf{V}\mathbf{W} - \mathbf{I})(\mathbf{X}^\mathsf{T} + \mathbf{s}\mathbf{1}^\mathsf{T} - (\bar{\mathbf{x}} - \mathbf{s})\mathbf{1}^\mathsf{T})\|_\mathrm{F}^2 \\ &= \|(\mathbf{V}\mathbf{W} - \mathbf{I})(\mathbf{X}^\mathsf{T} - \bar{\mathbf{x}}\mathbf{1}^\mathsf{T})\|_\mathrm{F}^2. \end{aligned}$$

Thus, we obtain the same value for the objective function using centered data:

$$f = \|\mathbf{W}\mathbf{V}^\mathsf{T} - \mathbf{X}_c + \mathbf{1}\boldsymbol{\delta}^\mathsf{T}\|_\mathrm{F}^2.$$

However, for centered data $\boldsymbol{\delta}^\mathsf{T} = \mathbf{0}$. Thus, the optimum value of the objective function is:

$$f = \|\mathbf{W}\mathbf{V}^\mathsf{T} - \mathbf{X}_c\|_\mathrm{F}^2,$$

which implies that we obtain the same optimum axis vectors (as in (27)) solving the optimization problems on centered data but discarding the term involving $\boldsymbol{\delta}$.

Optimal scaling of a single axis vector

Table. 2 shows the derivation of the solution to (20) for SC.

Gradient of the objective function in (22) for SC

Table. 3 shows the derivation of the gradient of the objective function (f) in (22) for SC.

The objective function in (20) for SC can be rewritten as follows:

$$
\begin{aligned}
P(\lambda) &= \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \lambda \mathbf{v}_n \end{bmatrix} \mathbf{p}_j - \mathbf{x}_j \right\|^2 = \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \lambda \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{V}} \\ \lambda \mathbf{v}_n \end{bmatrix}^{\mathsf{T}} \mathbf{x}_j - \mathbf{x}_j \right\|^2 = \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda \tilde{\mathbf{V}}\mathbf{v}_n \\ \lambda \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda^2 \mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n \end{bmatrix} \mathbf{x}_j - \mathbf{x}_j \right\|^2 \\
&= \sum_{j=1}^{N} \left\| \left(\begin{bmatrix} \tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda \tilde{\mathbf{V}}\mathbf{v}_n \\ \lambda \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda^2 \mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n \end{bmatrix} - \mathbf{I} \right) \mathbf{x}_j \right\|^2 = \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I} & \lambda \tilde{\mathbf{V}}\mathbf{v}_n \\ \lambda \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda^2 \mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n - 1 \end{bmatrix} \mathbf{x}_j \right\|^2 = \sum_{j=1}^{N} \mathbf{x}_j^{\mathsf{T}} \begin{bmatrix} \tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I} & \lambda \tilde{\mathbf{V}}\mathbf{v}_n \\ \lambda \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda^2 \mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n - 1 \end{bmatrix}^2 \mathbf{x}_j \\
&= \sum_{j=1}^{N} \begin{bmatrix} \tilde{\mathbf{x}}_j^{\mathsf{T}} & x_{j,n} \end{bmatrix} \begin{bmatrix} (\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I}) + \lambda^2 \tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})\tilde{\mathbf{V}}\mathbf{v}_n + \lambda^3 \tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n - \lambda \tilde{\mathbf{V}}\mathbf{v}_n \\ \lambda \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I}) + \lambda^3 \mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} - \lambda \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}} & \lambda^2 \mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n + \lambda^4 (\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n)^2 - 2\lambda^2 \mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n + 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_j \\ x_{j,n} \end{bmatrix} \\
&= \sum_{j=1}^{N} \Big(\tilde{\mathbf{x}}_j^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})\tilde{\mathbf{x}}_j + \lambda^2 \tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{x}}_j + \lambda \tilde{\mathbf{x}}_j^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})\tilde{\mathbf{V}}\mathbf{v}_n x_{j,n} + \lambda^3 \tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n} - \lambda \tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n \\
&\quad + \lambda x_{j,n}\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})\tilde{\mathbf{x}}_j + \lambda^3 x_{j,n}\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{x}}_j - \lambda x_{j,n}\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{x}}_j + \lambda^2 x_{j,n}\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n x_{j,n} + \lambda^4 (\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n})^2 - 2\lambda^2 x_{j,n}\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n} + x_{j,n}^2 \Big) \\
&= \sum_{j=1}^{N} \Big(\lambda^4 (\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n})^2 \; + \; 2\lambda^3 \tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n} \; + \; \lambda^2 \big(\tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{x}}_j + (\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n - 2\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n) x_{j,n}^2 \big) \\
&\quad + 2\lambda \tilde{\mathbf{x}}_j^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - 2\mathbf{I})\tilde{\mathbf{V}}\mathbf{v}_n x_{j,n} \; + \; \tilde{\mathbf{x}}_j^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - \mathbf{I})\tilde{\mathbf{x}}_j + x_{j,n}^2 \Big).
\end{aligned}
$$

Differentiating the polynomial yields:

$$
P'(\lambda) = 4\lambda^3 \sum_{j=1}^{N} (\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n})^2 \; + \; 6\lambda^2 \sum_{j=1}^{N} \tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n x_{j,n} \; + \; 2\lambda \sum_{j=1}^{N} \left(\tilde{\mathbf{x}}_j^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{x}}_j + (\mathbf{v}_n^{\mathsf{T}}\tilde{\mathbf{V}}^{\mathsf{T}}\tilde{\mathbf{V}}\mathbf{v}_n - 2\mathbf{v}_n^{\mathsf{T}}\mathbf{v}_n) x_{j,n}^2 \right) \; + \; 2 \sum_{j=1}^{N} \tilde{\mathbf{x}}_j^{\mathsf{T}}(\tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathsf{T}} - 2\mathbf{I})\tilde{\mathbf{V}}\mathbf{v}_n x_{j,n}.
$$

Table 2: *Derivation of the solution to (20) for SC.*

Firstly, the objective function in (22) for SC can be rewritten as follows:

$$f_{\mathrm{SC}}(\mathbf{v}_n) = \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \mathbf{v}_n^\mathsf{T} \end{bmatrix} \mathbf{p}_j - \mathbf{x}_j \right\|^2 = \sum_{j=1}^{N} \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \mathbf{v}_n^\mathsf{T} \end{bmatrix} [\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n]\mathbf{x}_j - \mathbf{x}_j \right\|^2 = \left\| \begin{bmatrix} \tilde{\mathbf{V}} \\ \mathbf{v}_n^\mathsf{T} \end{bmatrix} [\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n]\mathbf{X}^\mathsf{T} - \mathbf{X}^\mathsf{T} \right\|_{\mathrm{F}}^2 = \left\| \left(\begin{bmatrix} \tilde{\mathbf{V}} \\ \mathbf{v}_n^\mathsf{T} \end{bmatrix} [\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n] - \mathbf{I} \right) \mathbf{X}^\mathsf{T} \right\|_{\mathrm{F}}^2,$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. Furthermore, we can express $[\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n]$ as follows:

$$[\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n] = \mathbf{V}^\mathsf{T}\Delta + \mathbf{v}_n\zeta^\mathsf{T}, \tag{28}$$

where Δ is an $n \times n$ diagonal matrix whose entries are all 1, except its n-th component, which is 0. Also, ζ is a $n \times 1$ vector whose components are all 0, except its n-th entry, which is 1. We will use (28) to rewrite $f_{\mathrm{SC}}(\mathbf{v}_n)$ as follows:

$$f_{\mathrm{SC}}(\mathbf{v}_n) = \left\| \left[(\Delta\mathbf{V} + \zeta\mathbf{v}_n^\mathsf{T})(\mathbf{V}^\mathsf{T}\Delta + \mathbf{v}_n\zeta^\mathsf{T}) - \mathbf{I} \right] \mathbf{X}^\mathsf{T} \right\|_{\mathrm{F}}^2 = \left\| \left[\underbrace{(\Delta\mathbf{V}\mathbf{V}^\mathsf{T}\Delta + \Delta\mathbf{V}\mathbf{v}_n\zeta^\mathsf{T} + \zeta\mathbf{v}_n^\mathsf{T}\mathbf{V}^\mathsf{T}\Delta + \zeta\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\zeta^\mathsf{T})}_{\mathbf{E}} - \mathbf{I} \right] \mathbf{X}^\mathsf{T} \right\|_{\mathrm{F}}^2.$$

Expressing the Frobenius norm as a trace yields (note that matrix the $n \times n$ matrix $\mathbf{E}$ is symmetric):

$$f_{\mathrm{SC}}(\mathbf{v}_n) = \mathrm{tr}\left[\mathbf{X}(\mathbf{E} - \mathbf{I})^2\mathbf{X}^\mathsf{T}\right] = \mathrm{tr}\left[\mathbf{X}(\mathbf{E}^2 - 2\mathbf{E} + \mathbf{I})\mathbf{X}^\mathsf{T}\right] = \mathrm{tr}\left[\mathbf{X}\mathbf{E}^2\mathbf{X}^\mathsf{T}\right] - 2\mathrm{tr}\left[\mathbf{X}\mathbf{E}\mathbf{X}^\mathsf{T}\right] + \mathrm{tr}\left[\mathbf{X}\mathbf{X}^\mathsf{T}\right]. \tag{29}$$

The last term in (29) does not depend on $\mathbf{v}_n$ and is therefore irrelevant for the gradient. Thus, we will proceed by expanding the first two terms, using the following identities:

$$\Delta^2 = \Delta, \qquad \zeta^\mathsf{T}\zeta = 1, \qquad \Delta \cdot \zeta = \mathbf{0}, \qquad \mathbf{V}^\mathsf{T}\Delta\mathbf{V} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}, \qquad \mathbf{X}\Delta\mathbf{V} = \tilde{\mathbf{X}}\tilde{\mathbf{V}}, \qquad \text{and} \qquad \mathbf{X}\zeta = \mathbf{x}_n,$$

where $\tilde{\mathbf{X}}$ is the matrix composed of the first $n-1$ columns of $\mathbf{X}$, and $\mathbf{x}_n$ is the n-th column of $\mathbf{X}$.
Firstly,

$$\begin{aligned} -2\mathrm{tr}[\mathbf{X}\mathbf{E}\mathbf{X}^\mathsf{T}] &= -2\mathrm{tr}[\mathbf{X}\Delta\mathbf{V}\mathbf{V}^\mathsf{T}\Delta\mathbf{X}^\mathsf{T}] - 4\mathrm{tr}[\mathbf{X}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{V}^\mathsf{T}\Delta\mathbf{X}^\mathsf{T}] - 2\mathrm{tr}[\mathbf{X}\zeta\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\zeta^\mathsf{T}\mathbf{X}^\mathsf{T}] = -2\mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] - 4\mathrm{tr}[\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] - 2\mathrm{tr}[\mathbf{x}_n\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{x}_n^\mathsf{T}] \\ &= -2\mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] - 4\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\mathbf{x}_n] - 2\mathrm{tr}[\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n\mathbf{v}_n^\mathsf{T}] = -2\mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] - 4\mathbf{v}_n^\mathsf{T}\mathbf{a} - 2\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n, \end{aligned} \tag{30}$$

where $\mathbf{a} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\mathbf{x}_n$. Also, note that the first term does not depend on $\mathbf{v}_n$ and is therefore irrelevant for the gradient.
Secondly, we proceed by expanding $\mathrm{tr}[\mathbf{X}\mathbf{E}^2\mathbf{X}^\mathsf{T}]$. Since $\mathbf{E}$ has four terms, $\mathbf{E}^2$ has 16, but eight of them cancel due to $\Delta \cdot \zeta = \mathbf{0}$. Also, some terms appear twice. In particular, we have:

$$\begin{aligned} \mathrm{tr}[\mathbf{X}\mathbf{E}^2\mathbf{X}^\mathsf{T}] &= \mathrm{tr}[\mathbf{X}\Delta\mathbf{V}\mathbf{V}^\mathsf{T}\Delta^2\mathbf{V}\mathbf{V}^\mathsf{T}\Delta\mathbf{X}^\mathsf{T}] + 2\mathrm{tr}[\mathbf{X}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{V}^\mathsf{T}\Delta^2\mathbf{V}\mathbf{V}^\mathsf{T}\Delta\mathbf{X}^\mathsf{T}] + \mathrm{tr}[\mathbf{X}\Delta\mathbf{V}\mathbf{v}_n\zeta^\mathsf{T}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{V}^\mathsf{T}\Delta\mathbf{X}^\mathsf{T}] \\ &\quad + \mathrm{tr}[\zeta\mathbf{v}_n^\mathsf{T}\mathbf{V}^\mathsf{T}\Delta^2\mathbf{V}\mathbf{v}_n\zeta^\mathsf{T}] + 2\mathrm{tr}[\mathbf{X}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\zeta^\mathsf{T}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{V}^\mathsf{T}\Delta\mathbf{X}^\mathsf{T}] + \mathrm{tr}[\mathbf{X}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\zeta^\mathsf{T}\zeta\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\zeta^\mathsf{T}\mathbf{X}^\mathsf{T}] \\ = \ & \mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + 2\mathrm{tr}[\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + \mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + \mathrm{tr}[\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n\mathbf{x}_n^\mathsf{T}] + 2\mathrm{tr}[\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + \mathrm{tr}[\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{x}_n^\mathsf{T}] \\ = \ & \mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + 2\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\mathbf{x}_n] + \mathrm{tr}[\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\tilde{\mathbf{X}}\tilde{\mathbf{V}}\mathbf{v}_n] + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\mathbf{v}_n] + 2\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\mathbf{x}_n] + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n] \\ = \ & \mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + 2\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{D}\mathbf{a}] + \mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{C}\mathbf{v}_n] + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{D}\mathbf{v}_n] + 2\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{a}] + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathrm{tr}[\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n] \\ = \ & \mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + 2(\mathbf{v}_n^\mathsf{T}\mathbf{D}\mathbf{a}) + \mathbf{v}_n^\mathsf{T}\mathbf{C}\mathbf{v}_n + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n(\mathbf{v}_n^\mathsf{T}\mathbf{D}\mathbf{v}_n) + 2(\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{a}) + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n(\mathbf{v}_n^\mathsf{T}\mathbf{v}_n)(\mathbf{v}_n^\mathsf{T}\mathbf{v}_n), \end{aligned} \tag{31}$$

where $\mathbf{D} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}$, and $\mathbf{C} = \tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}\tilde{\mathbf{X}}\tilde{\mathbf{V}}$, which are both symmetric.
Substituting (30) and (31) in (29) we have:

$$\begin{aligned} f_{\mathrm{SC}}(\mathbf{v}_n) = & -4\mathbf{v}_n^\mathsf{T}\mathbf{a} - 2\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n + 2\mathbf{v}_n^\mathsf{T}\mathbf{D}\mathbf{a} + \mathbf{v}_n^\mathsf{T}\mathbf{C}\mathbf{v}_n + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\mathbf{D}\mathbf{v}_n + 2\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{a} + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n \\ & + \mathrm{tr}[\mathbf{X}\mathbf{X}^\mathsf{T}] - 2\mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}] + \mathrm{tr}[\tilde{\mathbf{X}}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\mathsf{T}\tilde{\mathbf{X}}^\mathsf{T}], \end{aligned}$$

where the terms involving traces do not depend on $\mathbf{v}_n$ and are therefore irrelevant for computing the gradient of the function.
Finally, taking the derivative with respect to $\mathbf{v}_n$ yields:

$$\begin{aligned} \nabla f_{\mathrm{SC}}(\mathbf{v}_n) &= -4\mathbf{a} - 4\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n + 2\mathbf{D}\mathbf{a} + 2\mathbf{C}\mathbf{v}_n + 2\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{D}\mathbf{v}_n + 4\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{a} + 2\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{a} + 4\mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\mathbf{v}_n \\ &= 2\mathbf{C}\mathbf{v}_n + 2\mathbf{v}_n^\mathsf{T}\mathbf{v}_n\mathbf{a} + \left(2\mathbf{D} - 4\mathbf{I} + 4\mathbf{v}_n\mathbf{v}_n^\mathsf{T}\right)\left(\mathbf{a} + \mathbf{x}_n^\mathsf{T}\mathbf{x}_n\mathbf{v}_n\right), \end{aligned}$$

where we have used the following rules:

$$\frac{\partial\mathbf{x}^\mathsf{T}\mathbf{b}}{\partial\mathbf{x}} = \mathbf{b}, \qquad \frac{\partial\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}}{\partial\mathbf{x}} = (\mathbf{A} + \mathbf{A}^\mathsf{T})\mathbf{x}, \qquad \frac{\partial\mathbf{x}^\mathsf{T}\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{b}}{\partial\mathbf{x}} = 4\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{b} + 2\mathbf{x}^\mathsf{T}\mathbf{x}\mathbf{b}, \qquad \text{and} \qquad \frac{\partial\mathbf{x}^\mathsf{T}\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{x}}{\partial\mathbf{x}} = 4\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{x},$$

where $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, and $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Table 3: *Derivation of the gradient of the objective function in (22) for SC.*