Burning Sage: Reversing the Curse of Dimensionality in the Visualization of High-Dimensional Data

Ursula Laa, Dianne Cook, and Stuart Lee

School of Physics and Astronomy, Monash University, Melbourne, Australia; Department of Econometrics and Business Statistics, Monash University, Melbourne, Australia; Institute of Statistics, University of Natural Resources and Life Sciences, Vienna, Austria; Molecular Medicine Division, Walter and Eliza Hall Institute, Parkville, Australia

ABSTRACT
In high-dimensional data analysis, the curse of dimensionality reasons that points tend to be far away from the center of the distribution and on the edge of high-dimensional space. Contrary to this, is that projected data tends to clump at the center. This gives a sense that any structure near the center of the projection is obscured, whether this is true or not. A geometric transformation to reverse the curse, is defined in this article, which uses radial transformations on the projected data. It is integrated seamlessly into the grand tour algorithm, and we have called it a burning sage tour, to indicate that it reverses the curse. The work is implemented into the tourr package in R. Several case studies are included that show how the sage visualizations enhance exploratory clustering and classification problems. Supplementary files for this article are available online.

1. Introduction

The term “curse of dimensionality” was originally introduced by Bellman (1961), to express the difficulty of doing optimization in high dimensions because of the exponential growth in space as dimension increases. A way to think about it is, that the volume of the space grows exponentially with dimension, which makes it infeasible to sample enough points—any sample will be less densely covering the space as dimension increases. The effect is that most points will be far from the sample mean, on the edge of the sample space. Hall, Marron, and Neeman (2005) have shown that in the extreme case of high-dimension, low-sample size data, observations are on the vertices of a simplex.

This affects many aspects of data analysis: minimizing the error during model fitting relies on effective optimization techniques, nonparametric modeling requires finding nearest neighbors which may be far away and sampling from high-dimensional distributions is likely to have points far from the population mean. Although not directly relevant for the topic of this article, it is interesting to note that Donoho (2000) considered the curse of dimensionality as a blessing, because the sparsity can be leveraged for computational efficiency. This is used in regularization methods, like lasso, to penalize model complexity. The penalty term results in shrinking (some of) the parameter estimates toward zero.

Paradoxically, the curse of dimensionality inverts for dimension reduction, resulting in an excessive amount of observations near the center of the distribution. This affects visualizations made on low-dimensional projections. The effect is described by Diaconis and Freedman (1984), showing that most linear projections onto low-dimensional subspaces are approximately Gaussian, and that observations concentrate near the center as the dimensionality of the full space increases. This has motivated the development of indexes for projection pursuit which search for departure from normality. It is also related to what is called “data piling” in high-dimensional low-sample size data (Marron, Todd, and Ahn 2007; Ahn and Marron 2010): all observations can collapse into a single point. These issues also persist with nonlinear dimension reduction techniques, and are often referred to as the “crowding problem,” which methods like t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton 2008) aim to alleviate.

Here, we are providing a solution to fix a long-standing problem in the tour (Asimov 1985; Buja et al. 2005), where high-dimensional data are viewed by a movie of low-dimensional projections. Points crowding the center of the drawing canvas detract from reading the plots, even with relatively few dimensions. This is a structural problem related purely to the geometry of projections from high-dimensional spaces. We show that it can be alleviated with a radial transformation on the low-dimensional space. Figure 1 illustrates this crowding problem, as would be experienced when using a tour. Two-dimensional linear projections of data, generated by sampling uniformly within \( p \)-dimensional hyperspheres (\( p = 3, 10, 100 \)), are displayed as hexbin plots. Color indicates log count of the bin, with yellow mapping the highest counts. As \( p \) increases the density concentrates in the center of the projection. (Note that, sampling from a uniform in a sphere is used for this illustration for several reasons. Any linear projection is equivalent because of the rotation invariance of the distribution. The uniform sampling
places focus on the space, points are spread evenly across the volume, and not a data distribution. Also, \( p = 10 \) would already be considered high-dimensional, as seen by the piling observed in the middle plot.)

Common solutions to overplotting include transparency and binned or density displays (Carr et al. 1987), for a practical and accessible discussion, see (Chang 2018, sec. 5.5). These are designed to tackle large sample size, rather than dimension reduction problems. A low-dimensional projection represented as a scatterplot, binned plot, or contour plot, will still suffer from data piling and reduced resolution near the center of the projection. These approaches are still useful for large sample sizes, and could be applied after the transformation.

The underlying idea of the (radial) transformation, to reverse the curse, is to correct for the distortion in geometry that occurs when projecting from a high-dimensional space onto low dimensions. This is done by mapping equal volume in the full space onto equal area in a two-dimensional projection via a geometrical transformation of the projected data. When used with data that is uniformly distributed in the high-dimensional space, the transformation will result in a uniform distribution in the low-dimensional display. In general, for any underlying data distribution, we can instead think about the effect in terms of resolution: adjusting for the geometric piling effect means that the resolution depends on the projected volume. The observed projected distribution is representative of the original high-dimensional distribution in this sense.

The article is structured as follows. The radial transformation and its implementation is described in Section 2. Section 3 illustrates the use of the sage tour with examples in clustering, supervised classification and a classical needle-in-the-haystack problem. Section 4 describes possible extensions to the method.

2. Burning Sage Algorithm

To understand why points tend to be away from the center in the high-dimensional space, but crowd the center in low-dimensional projections, it is helpful to consider the projected volume relative to high-dimensional volume. To avoid edge effects and to impose rotation invariance, we start from the volume in a hypersphere, that is, the volume is contained within a specified distance from the center. This is more tractable than assuming a hypercube (box).

Figure 2 illustrates the comparison to be made, using something we can easily picture, a 3D sphere. Projecting the data from within a 3D sphere to 2D (gray disk) will result in mass being condensed into the disk. Imagine comparing the volume of a cylinder at different locations in the disk. A centered cylinder has more volume. This is exaggerated as \( p \) increases: the centered cylinder has much more volume than any other cylinder.

To reverse this effect, we introduce a radial transformation that redistributes the projected points, such that equal volume in the original (\( p \)-dimensional) space is projected onto equal area in a two-dimensional projection. Note that this can be generalized for \( d \)-dimensional projections by mapping onto equal \( d \)-dimensional volume instead.

2.1. Definition of the Relative Projected Volume

To understand how the \( p \)-dimensional volume is projected onto a two-dimensional plane, we study what fraction of the total volume is projected onto the area of a disk depending on its radius. This dependence was described in Laa et al. (2020). We start from a \( p \)-dimensional hypersphere, with radius \( R \) and volume \( V(R,p) \), and its projected volume onto a centered two-dimensional disk of radius \( r \), \( V_{\text{2D}}(r,p,R) \), where \( r \) can be any radius within \([0,R]\). The relative projected volume is then given as the ratio of these two quantities,

\[
v_2(r;p,R) = \frac{V_{\text{2D}}(r,p,R)}{V(R,p)} = 1 - \left(1 - \left(\frac{r}{R}\right)^2\right)^{p/2}. \tag{1}
\]

We can compare \( v_2(r;p,R) \) to the relative volume within a radius \( r \) in the original \( p \)-dimensional hypersphere,

\[
v_p(r;p,R) = \frac{V(r,p)}{V(R,p)} = \left(\frac{r}{R}\right)^p. \tag{2}
\]

Figure 3 compares these two quantities (Eqs. (1) and (2)), for \( p = 3,10, \) and 100. On the left is \( v_2(r;p,R) \) and on the right
Figure 2. Illustration (and notation) for describing the elements used in the burning sage transformation. The 3D sphere (left) shows the different volumes to be compared. The full sphere has volume $V(R, p)$. Within a radius $r$ the sphere contains the reduced volume $V(r; p, R)$, shown in blue, but the projected volume within a radius $r$ in a two-dimensional plane is much larger, given by the volume of the cylinder with rounded caps, $V_{2D}(r; p, R)$, shown in red. The intersection of the plane with the sphere is illustrated in gray, and the plane representing the projection with both radii is shown at right.

Figure 3. Comparing relative volume of a $p$-dimensional hypersphere captured within a radius $r$, in the 2-dimensional projection (left) and in the $p$-dimensional space (right), for $p = 3, 10, \text{ and } 100$. The difference is dramatic, which illustrates the paradox of the curse of dimensionality. The relative volume of a $pD$ sphere shrinks, as $p$ increases, while the projected volume (near the center) grows.

is $V_p(r; p, R)$. The function shapes change in opposite directions as $p$ increases: $V_2(r; p, R)$ peaks earlier and $V_p(r; p, R)$ gets flatter. This is the paradox of the curse of dimensionality in that projected volume at the center increases with $p$.

2.2. Calculating the Radial Transformation

The aim of the algorithm is to redistribute the projected volume such that equal relative areas on the disk, as given by $V_2(r; p = 2, R) = V_p(r; p = 2, R) = (r/R)^2$, contain equal relative projected volume, given by $V_2(r; p, R)$. This is achieved through a transformation of the projected radius that can be defined for any $r \in [0, R]$, and is applied to the projected data points in the plane, $y = (y_1, y_2)$. We work with polar coordinates and represent the data points as $y = (r_y, \theta_y)$. The angular component $\theta_y$ is uniform for this distribution, by the rotation invariance of the sphere, and thus does not need to be transformed.

The radial component $r_y$ is transformed in two steps. First, we replace it with $V_2(r_y; p, R)$ to redistribute the volume uniformly by the radius. The second step is then to transform $V_2(r_y; p, R)$ using the inverse of $V_2(r_y; 2, R)$, to go from a uniform distribution in radius to a uniform distribution in area of the disk. This inverse is defined via $V_2^{-1}(V_2(r_y; 2, R); 2, R) = V_2^{-1}(V_2^{-1}(r_y; 2, R); 2, R) = r_y$ and thus $V_2^{-1}(r_y; 2, R) = R\sqrt{r_y}$.

The full radial transformation is therefore given by

$$r'_y = V_2^{-1}(V_2(r_y; p; R); 2, R) = R\sqrt{1 - \left(1 - \left(\frac{r_y}{R}\right)^2\right)^{p/2}}.$$ (3)

The relation between $r'_y$ and $r_y$ depends on the number of dimensions $p$, and is illustrated for selected values in Figure 4. We see that the transformation is approximately linear near the center. As $p$ increases it becomes nonlinear faster, and for $p =$
10, for example, the points with radius \( r_y > 0.5 \) will already be highly distorted and pushed out toward the last eighth in \( r'_y \).

Figure 5 demonstrates this for different values of \( p \) by showing equidistant circles for which the radius has been transformed according to Equation (3).

At this point a few characteristics of the transformation should be noted. First, as already mentioned, it is approximately linear near the center, and thus will not affect shapes in that part of the graph. In addition, since only the radial component is transformed, angles between the points will not be altered. (This can also be shown explicitly, from mathematical reformulation of the angles between two points after the transformation.) Note however that shapes will be affected further away from the center (in particular toward the maximum range \( R \)). For example, a linear relation might appear curved because of the nonlinear nature of the transformation in that region of the projected space. Thus, while the interpretation is straightforward for points projected in the central region, care must be taken with the interpretation of features observed in the outer region of the projected distribution.

2.3. Trimming and Tuning

The transformation in Equation (3) is fixed for a given input dataset, by evaluating the number of dimensions \( p \) and the maximum distance from the center \( R \). However, in practice we may wish to trim the projected data or tune the transformation. A combination of both adjustments can be used to further zoom in on the center of the distribution, or alternatively, to soften the transformation.

2.3.1. Trimming

The overall scale of the transformation is determined by \( R \). In the case of an approximately spherical and uniform distribution, the maximum distance from the center works well and ensures the validity of the rescaling in Equation (3). But this is not robust and might result in a much larger scale than desired, especially when it is determined by outlying observations.

We therefore allow trimming of the projected observations, using \( R \) as a free parameter of the display function. When selecting a value \( R \) that is smaller than the maximum distance from the center, we need to ensure that the projected radius of points is always smaller than \( R \), by trimming \( r_y \) as

\[
    r^{\text{trim}}_y = \min(r_y, R)
\]

for each observation.

2.3.2. Tuning

The dimension of the input might not reflect the intrinsic dimensionality of the dataset. This could be the case when dimension reduction was used prior to visualization, for example, displaying only the first few principal components. In this case, the effective dimensionality \( p_{\text{eff}} \) is likely between the original number of dimensions and the selected number of principal components. We can think of omitted components as being in the orthogonal space of all considered projections, with some directions being pure noise, while others may still carry relevant information.

We allow tuning \( p_{\text{eff}} = \gamma p \) by selecting the scaling parameter \( \gamma \). By default, \( \gamma = 1 \) and \( p_{\text{eff}} = p \). When \( \gamma < 1 \) the rescaling will be softer, and \( \gamma > 1 \) results in more aggressive rescaling than suggested by \( p \) alone. Note that when \( p_{\text{eff}} < 2 \) we actually invert the behavior and shift the focus away from the center, in general this is not recommended.

2.4. Data Preprocessing

Before applying the transformation, it can be important to preprocess the data, as is generally required with methods for the visualization of high-dimensional data. In particular, when working with projections that combine multiple variables it is important to ensure a common scale across the dimensions. This is especially important with the sage display, where the overall scale (set by the maximum radius \( R \)) determines the effect of the transformation. Scaling is commonly also combined with centering as a preprocessing step. This is less important in our implementation because we always center the projected data before applying our transformation (i.e., the projected mean defines the origin of our polar coordinate system in the plane). Centering and scaling together would be achieved by standardizing each variable in the data to be centered at 0 and have a variance equal to 1.

We may also wish to apply nonlinear transformations before visualizing the data or combining variables in a projection. One example would be to use a log transformation in the case
of a variable that has a skewed distribution. These types of transformations will generally change the distribution across the high-dimensional volume (and in particular also the position of the mean), and can thus make the sage display more useful for a wide range of observed data distributions.

The third common preprocessing would involve a dimension reduction step. Even with a relatively low number of dimensions, for example, with $p = 10$, we see noticeable piling. However, we are often confronted with much higher dimensional datasets in which many of the variables are primarily adding noise. In this case, it is advisable to first reduce the number of dimensions, for example using principal components analysis. Other dimension reduction techniques could be used. The sage tour would then be used to view the first few components, which would likely provide more insight about the data structures than is possible from only the first two.

2.5. Implementation as a Dynamic Display

While the radial transformation can in general be used with any low-dimensional display that suffers from data crowding, it is most useful when combined with a dynamic display showing a sequence of interpolated low-dimensional projections obtained when running a tour. We have implemented it as a new display method called `display_sage` in the `tourr` package (Wickham et al. 2011) in R (R Core Team 2020).

We can think of the display functions as part of a data pipeline obtained when running a tour. The initial step is preprocessing the data, given by $X$, an $n \times p$ matrix containing $n$ observations in $p$ dimensions. Typically, this includes centering and scaling, using either the overall range or the variance. Ensuring a common scale of all variables, comparable to the selected scale parameter $R$, is especially important with the new display. The tour then iterates over the following steps:

1. Obtain projection matrix $A$. For $d$-dimensional projections this is an orthonormal $p \times d$ matrix. To ensure the smooth rotation of projections, each new $A$ is obtained as an interpolated step in the sequence, as explained in Buja et al. (2005).
2. Project the data by computing $Y = X \cdot A$.
3. Map $Y$ to the display to re-draw the projected data. For $d = 2$ this typically maps the projected points onto a scatter-plot display. With the new display we first transform $Y$ as follows:
   - Center the two-dimensional matrix $Y$ and compute its polar coordinate representation $(r_y, \theta_y)$.
   - For each observation, first use Equation (4) to get the trimmed radius $r_{trim}'$ within the specified range, and then apply the radial transformation defined in Equation (3) to obtain $r_y'$.
   - Use the transformed radial coordinate $r_y'$ to recompute the mapping onto Euclidean coordinates $(y_1', y_2')$.
4. To fit the projected and transformed data $Y'$ onto the plotting canvas ranging between $[-1, 1]$, we rescale each mapped observation $y$ using a scaling parameter $s$.

The display can be added when calling the `animate` function in `tourr`, as

```r
tourr::animate(data, tour path = tourr::grand tour(), display = display_sage(gam, R, half range))
```

and uses `gam` to set the $y$ parameter for computing $p_{eff}$, and the overall range $R$ used for trimming. Both these parameters are described in Section 2.3.

For consistency with other display functions in the `tourr` package, we also pass in the scale parameter $s$ via the `half_range` argument. By default this is evaluated from the range of the data, but kept as a free parameter to zoom in (without any transformation) if necessary. This can be important, for example, when the range is large because of an outlying point. Note here that the ratio $R/s$ sets the scale for fitting the displayed data on the plotting canvas, by default $s = R$. When adjusting $R$ the user should take care to adjust $s$ accordingly.

3. Applications

To illustrate the benefit of using the reverse transformation for examining data using a tour, four applications are shown: clustering of single cell RNA-seq, classifying hand-sketched images, comparing physics experiments, and the classical pollen data. The pollen data example is used to illustrate the effect of parameter choices in the sage tour.

3.1. Clustering Single-Cell RNA-seq Datasets

In the analysis of single cell RNA-seq data, cluster analysis is performed to detect cell types and characterize the expression of genes that define those cell types, and the relative orientation of the cell types to each other (trajectory analysis) (Amezquita et al. 2020). Generally, for cluster verification, analysts use embedding methods like t-SNE to verify the placement and meaning of clusters from a clustering algorithm. An alternative is to use a tour on a small number of principal components to examine the clusters relative to gene expression.

Here, we compare the sage display and regular tour display on mouse retinal single cell RNA-seq data from Macosko et al. (2015). The raw data consist of a 49,300 cells and was downloaded using the `scRNAseq` Bioconductor package (Risso and Cole 2019). We use a standard workflow for preprocessing and normalizing this data (described by Amezquita et al. 2020): quality control was performed using the `scater` package (McCarthy et al. 2017) and `scran` (Lun, McCarthy, and Marioni 2016) was used to transform and normalize the expression values and select highly variable genes (HVGs). The top 10% of the most HVGs were used as features to subset the normalized expression matrix and compute the principal components. Using the first 25 PCs we built a shared nearest neighbors graph (with $k = 10$) and clustered this graph using Louvain clustering, resulting in 11 clusters being formed (Blon-del et al. 2008).

A tour is run on the first five PCs (approximately 20% of the variance in expression), on a weighted subsample of cells based on their cluster membership—4590 cells. For the sage display,
we set $\gamma = 3$, fixing the effective dimensionality of the data to $p_{\text{eff}} = 15$. The PCs are scaled to have zero mean and unit variance. Here, we focus on comparing three of the clusters. In the PC plots they look very similar, begging the question whether they should be considered to be separate groups. Figure 6 shows selected frames from a default tour (top row) and the sage tour (bottom row). The columns show the same projection, with the difference being that the sage transformation is applied in the sage tour projections. The full animations are available in the supplementary material. The static plots serve to illustrate the main points, but we encourage the reader to look at the tour animations to fully appreciate the advantage of the sage display.

Using the default tour display (Figure 6, top), the three clusters (dark green, blue, and yellow) are obscured by points in other clusters as we move through the frames of the animation. The points in the dark green cluster are overlapping those found in the yellow and blue clusters; and it is difficult to see if there is any separation between the blue and yellow clusters. In contrast, the sage display (Figure 6, bottom), expands the center of projection, and results in the differences between the three clusters being more visible. Particularly, the relative positions of the yellow and blue clusters are easier to see. While these clusters are distinct from the dark green cluster, in the most frames they are still overlapping and mixed together, providing evidence that it may be appropriate to consider them a single cluster. Conversely, it can be seen that the dark green cluster is distinctly separated from the other two in some projections. The sage tour makes these comparisons a little easier.

### 3.2. Classifying Hand-Sketches

We next use the new display to look at different distributions of images from the Google QuickDraw collection (Google, Inc 2020). These are $28 \times 28 = 784$ pixel gray scale data that are available publicly. In this example, we sample 1000 images from three types of sketches (banana, cactus, crab) and see if we can separate the classes in the high-dimensional parameter space.

We reduce the dimensionality from 784 variables to the first 5 PCs, which captures approximately 20% of the variation of the data. Before applying the tour, we rescale each component to have mean zero and unit variance. To account for the dimension reduction before visualization, we set $\gamma = 2$ for the sage display.

Figure 7 shows the grand tour on the PCs, where green points correspond to the banana class, orange points represent the cactus class and purple points are the crab class. In the selected frames of both displays points belonging to the cactus class are concentrated near the center, however on the default display (Figure 7, top), there is overplotting: points from other classes overlap those in the cactus class. The sage display (Figure 7, bottom) helps reduce overplotting, it is easier to see that the centers of class are separated and that there is substructure in the banana class, which further collapses into two subgroups.

The animated sage tour available in the supplementary material further reveals a low density of points near the center of the distribution: observing the movement of points when rotating the viewing angle shows that even the cactus class is clustering away from the mean.

### 3.3. Comparing Physics Experiments: PDFSense

Data were obtained from CT14HERA2 parton distribution function fits and describe the sensitivity of fit parameters to experimental measurements (Wang et al. 2018). There are 28 parameters, and varying one at a time to move $\pm \sigma$ away
Figure 7. Selected frames of the tour run on the sketches data using the default tour display (top), and using the sage display with $\gamma = 2$ (bottom). Three types of sketches are indicated by color: banana (green), cactus (orange) and crab (purple). Overplotting of points is a problem for the grand tour display, while the sage display reveals low density near the center.

Figure 8. Selected frames of the tour of the pdfsense data using the default tour display (top), and using the sage display with $R = 10$ (bottom). Different underlying physical processes are shown by color and we can see orthogonality between the three groups. The sage display preserves the overall structure while revealing details that are hidden near the center in the default display.

from the “best fit point” (maximum likelihood estimate) provides our input variables, labeled X1–X56. Each of the 2808 observations corresponds to a physical observable and measures how the fit prediction changes along the 56 directions in parameter space. Points are grouped based on the underlying process in the experiment, which is mapped to color in the following. With the analysis of the distribution along these variables X1–X56 we can understand to what extent each experimental measurement provides new information for the global fit. For example, orthogonality between groups marks complementary constraints, and outlying points are considered as important for future fits, see discussion in Cook, Laa, and Valencia (2018).

Following the processing described there, we tour the first 6 PCs, rescaled to have zero mean and unit variance. In Figure 8 we see that the sage display with $R = 10$ (Figure 8,
Figure 9. Selected views of the pollen data in the new sage display, with default settings (left), setting $R = 1$ (middle) and $\gamma = 20$ (right). We can tune either $\gamma$, $R$ or a combination of the two to reveal the word “EUREKA” near the center of the distribution.

bottom), maintains the overall shape of the data seen using the default tour display (Figure 8, top). The different physical process, shown in different colors, are indeed orthogonal in the parameter space, as can be seen most clearly by looking at the animations available in the supplementary material.

The particular structure of this distribution, with some clusters extending linearly away from the center and a set of outlying points, results in poor use of the plotting space, and high level of clustering near the center. For example, focusing on the blue cluster, we can see that it extends out along different directions, but it can be challenging to observe how the points move under the tour rotation, as overplotting becomes an issue when points move through the center. Here, the new display (bottom row) shows a clearer view.

3.4. Tuning the Parameters: Pollen

The classical pollen data are useful to demonstrate the trimming and tuning parameters. The five-dimensional data set was simulated by David Coleman of RCA Labs, for the Joint Statistics Meetings 1986 Data Expo (Coleman 1986), and is an example of a hidden structure near the center of a distribution. The data are standardized by centering and scaling such that the standard deviation of each variable is equal to one.

Neither the standard tour display nor the sage display with default settings ($R = 6.6$ which is set by the data scale, and $\gamma = 1$) reveals the structure (left plot in Figure 9). We can use either $\gamma$, $R$ or a combination of the two to zoom in further near the center. For example, we can use trimming ($R = 1, \gamma = 1$) (middle plot) or tuning ($R = 6.6, \gamma = 20$) (right plot) as shown in Figure 9. There is an approximate equivalence between the results obtained using either tuning or trimming, and both views clearly reveal the word “EUREKA” hidden in the distribution.

While the static views look very similar, comparing the tour animations (available in the supplementary material) reveals some differences between the display with trimming or tuning. When trimming (by setting $R = 1$) the focus is clearly on the center of the distribution, and most points get pushed out toward a maximum radius circle. On the other hand, tuning the display by setting $\gamma = 20$ preserves the elliptical shape of the distribution, making it easier to see correlation patterns.

4. Discussion

This article has introduced the sage tour, which reduces the data crowding effects that occur when taking low-dimensional projections of high-dimensional data. This new technique is easily incorporated into exploratory high-dimensional data analysis, and applications shown in Section 3 provide examples of the following tasks:

- Clustering: the sage display uncovered clusters that were originally obscured by data piling, while still giving the viewer an accurate assessment of the size of a cluster, and their relative orientation, as shown in the single cell RNA-seq example (Section 3.1)
- Classification: the sage display decreases the number of overlapping points between classes and provides better visual separation between classes compared to the regular tour, as shown in the sketches example (Section 3.2)
- Shape analysis: the sage display helps us understand structures across multiple dimensions, for example orthogonality between multiple groups, as shown in the pdfsense example (Section 3.3)
- Needle discovery: the sage display allows to find hidden signal that is concealed by the density of points around the center of the projection, as shown in the pollen example (Section 3.4)

The approach provides interpretable visualization that captures high dimensional information and preserves global structure, and it is complementary to nonlinear dimension reduction techniques. For example, when visualizing clusters, the sage display enables an assessment of cluster shapes, and accurately captures relative position and orientation. The burning sage transformation is global and does not magnify local structure like t-SNE does.

Note here that adjusting the resolution using the burning sage transformation will lead to an overall decrease in density near the center. When studying the separation between groups it is important to keep in mind that the effect is the same within and between clusters. Thus, if points within a cluster remain grouped while separation between clusters becomes more apparent, this can be interpreted as a true finding. Conversely, in the case of a visual artifact, we expect a general reduction of density that does not lead to better separation between the clusters.
We have motivated the transformation from geometric arguments: equal volume of the high-dimensional hypersphere should be mapped onto equal area in the two-dimensional projection plane. We could also think about this is in terms of the distribution function: a uniform distribution in high-dimensions should still look uniform in the projected view. Our approach is defining the transformation assuming that points are uniformly distributed in the hypersphere. This could be generalized to work with the empirical radial cumulative distribution function instead.

An alternative is the slice tour (Laa, Cook, and Valencia 2020) which allows distributions of points around the center of the data to be explored using sections instead of projections. Instead of adjusting for effects from projecting a high-dimensional volume, the solution with the slice tour is to only highlight projected points if they are within a small volume around the current projection plane, thus showing a local view of the distribution. The slice tour is useful when there are large numbers of observations or if there is concave structure in the data. Note that only few methods are currently available for looking at distributions. Theslicetourisusefulwhentherearelargenumbersof observations or if there is concave structure in the data. Note that only few methods are currently available for looking at sections of the data in a systematic fashion, and further work is needed to fully explore the potential of this type of visualization. In smaller datasets we may also use interactive approaches such as linked brushing to explore for local structures or conditional features.

The tuning parameters can be used to more aggressively expand the center of the display. All of the examples shown had some tuning. The last example demonstrated how points away from the projected center get moved to the edge of the hypersphere as $γ$ is increased or $R$ is decreased. With more center magnification, the nonlinear transformation can introduce distortions, but this is a well known problem for any nonlinear dimension reduction technique including t-SNE. However, unlike t-SNE, any distortion introduced by the slice tour is interpretable because it is controlled by a simple function Equation (3).

The sage display is fast to compute, which lends itself to being embedded into an interactive interface. An ideal interface would allow real time changes to the parameters of the transformation. This would be especially useful when coupled with linked brushing in complementary views.

Supplementary Material

The source material for this article is available at https://github.com/uschiLaa/burning-sage. The animated gifs for all applications are also included in html files in the supplementary material.

Acknowledgments

While the article title was originally derived from a pun, we thank the reviewer who suggested that it could also be considered an acronym: BuRnNG SaGe: Baseline Radial Non-Linear Geometric Scaling for Graphical Exploration. The article was written in \texttt{markdown} (Xie, Allaire, and Grolemund 2018) using \texttt{knitr} (Xie 2015).

Funding

The authors gratefully acknowledge the support of the Australian Research Council (DP170103884).

ORCID

Ursula Laa http://orcid.org/0000-0002-0249-6439
Dianne Cook http://orcid.org/0000-0002-3813-7155

References

Ahn, J., and Marron, J. S. (2010), "The Maximal Data Piling Direction for Discrimination," Biometrika, 97, 254–259. [40]
Amezquita, R. A., Lan, A. T. L., Becht, E., Carey, V. J., Carpp, L. N., Getzinger, L., Marini, F., Rue-Albrecht, K., Risso, D., Soneson, C., Waldron, L., Pagès, Hervé, Smith, M. L., Huber, W., Morgan, M., Gottardo, R., and Hicks, S. C. (2020), "Orchestrating Single-Cell Analysis With Bioconductor," Nature Methods, 17, 137–145. [44]
Asimov, D. 1985. "The Grand Tour: A Tool for Viewing Multidimensional Data," SIAM Journal of Scientific and Statistical Computing, 6 (1), 128–143. [40]
Bellman, R. (1961), Adaptive Control Processes : A Guided Tour, Princeton Legacy Library, Princeton, NJ: Princeton University Press. [40]
Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008), "Fast Unfolding of Communities in Large Networks," Journal of Statistical Mechanics, P10008. [44]
Buja, A., Cook, D., Asimov, D., and Hurley, C. (2005), "14—Computational Methods for High-Dimensional Rotations in Data Visualization," in Data Mining and Data Visualization, Vol. 24 of Handbook of Statistics, eds. C.R. Rao, E.J. Wegman and J.L. Solka, 391–413, Amsterdam: Elsevier. [40,44]
Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Littlefield, J. S. (1987), “Scatterplot Matrix Techniques for Large N,” Journal of the American Statistical Association, 82, 424–436. [41]
Chang, W. (2018), R Graphics Cookbook: Practical Recipes for Visualizing Data (2nd ed.), O'Reilly Media. Available at https://r-graphs.org. [41]
Coleman, D. 1986. "Geometric Features of Pollen Grains." Available at http://lib.stat.cmu.edu/data-expo/. [47]
Cook, D., Laa, U., and Valencia, G. (2018), “Dynamical Projections for the Visualization of PfSense Data,” European Physical Journal C, 78, 742. [46]
Daconis, P., and Freedman, D. (1984), “Asymptotics of Graphical Projection Pursuit,” Annals of Statistics, 12, 793–815. https://doi.org/10.1214/aos/1176346703. [40]
Donoho, D. L. (2000), “High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality.” Unpublished paper presented at AMS Conference on Math Challenges of the 21st Century. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.329.3392. [40]
Google, Inc. 2020. “Quick, Draw! The Data.” Available at https://quickdraw.withgoogle.com/data. [45]
Hall, P., Marron, J. S., and Neeman, A. (2005), "Geometric Representation of High Dimension, Low Sample Size Data," Journal of the Royal Statistical Society, Series B, 67, 427–444. [40]
Laa, U., Cook, D., Buja, A., and Valencia, G. (2020), "Hole or Grains? A Section Pursuit Index for Finding Hidden Structure in Multiple Dimensions," arXiv:2004.13327. [41]
Laa, U., Cook, D., and Valencia, G. (2020), "A Slice Tour for Finding Hollowness in High-Dimensional Data," Journal of Computational and Graphical Statistics, 1–10. [48]
Lun, A. T. L., McCarthy, D. J., and Marioni, J. C. (2016), "A Step-by-Step Workflow for Low-Level Analysis of Single-Cell RNA-seq Data With Bioconductor," F1000Research, 5, 2122. [44]
Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tiresh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Soneson, J. R., Shakle, A. K., Regev, A., McCarthy, S. A. (2015), "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets," Cell, 161, 1202–1214. [44]
Marron, J. S., Todd, M. J., and Ahn, J. (2007), "Distance-Weighted Discrimination," Journal of the American Statistical Association, 102, 1267–1271. [40]
McCarthy, D. J., Campbell, K. R., Lan, A. T. L., and Willis, Q. F. (2017), "Scatter: Pre-Processing, Quality Control, Normalisation and Visualisation of Single-Cell RNA-seq Data in R,“ Bioinformatics, 33, 1179–1186. [44]
R Core Team. (2020), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. [44]

Risso, D., and Cole, M. (2019), *scRNAseq: Collection of Public Single-Cell RNA-Seq Datasets*, R package version 2.0.2. [44]

van der Maaten, L., and G. Hinton. 2008. "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, 9, 2579–2605. [40]

Wang, B.-T., Hobbs, T. J., Doyle, S., Gao, J., Hou, T.-J., Nadolsky, P. M., and Olness, F. I. (2018), "Mapping the Sensitivity of Hadronic Experiments to Nucleon Structure," *Physical Review D*, 98, 094030. [45]

Wickham, H., Cook, D., Hofmann, H., and Buja. 2011. "tourr: An R Package for Exploring Multivariate Data with Projections," *Journal of Statistical Software*, 40, 1–18. http://www.jstatsoft.org/v40/i02/. [44]

Xie, Y. (2015), *Dynamic Documents With R and knitr* (2nd ed.), Boca Raton, FL: Chapman and Hall/CRC. [48]

Xie, Y., Allaire, J. J., and Grolemund, G. (2018), *R Markdown: The Definitive Guide*, London: Chapman and Hall/CRC. [48]