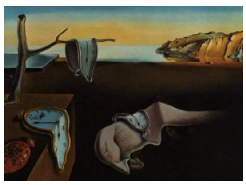

Journal of Statistical Software
July 2012, Volume 50, Issue 2. http://www.jstatsoft.org/

Data Analysis with the Morse-Smale Complex: The msr Package for R

Samuel Gerber
University of Utah

Kristin Potter
University of Utah

Abstract

In many areas, scientists deal with increasingly high-dimensional data sets. An important aspect for these scientists is to gain a qualitative understanding of the process or system from which the data is gathered. Often, both input variables and an outcome are observed and the data can be characterized as a sample from a high-dimensional scalar function. This work presents the R package **msr** for exploratory data analysis of multivariate scalar functions based on the Morse-Smale complex. The Morse-Smale complex provides a topologically meaningful decomposition of the domain. The **msr** package implements a discrete approximation of the Morse-Smale complex for data sets. In previous work this approximation has been exploited for visualization and partition-based regression, which are both supported in the **msr** package. The visualization combines the Morse-Smale complex with dimension-reduction techniques for a visual summary representation that serves as a guide for interactive exploration of the high-dimensional function. In a similar fashion, the regression employs a combination of linear models based on the Morse-Smale decomposition of the domain. This regression approach yields topologically accurate estimates and facilitates interpretation of general trends and statistical comparisons between partitions. In this manner, the **msr** package supports high-dimensional data understanding and exploration through the Morse-Smale complex.

Keywords: Morse-Smale complex, visualization, exploratory data analysis, regression, high-dimensional data.

1. Introduction

Recent advances in computational topology have led to a multitude of algorithms to estimate the topological properties of a function from a finite sample (Van Kreveld *et al.* 1997; Edelsbrunner *et al.* 2003; Carr *et al.* 2003; Gyulassy *et al.* 2007). These algorithms led to the analysis of data sets based on their topological properties and have shown promising results

in a wide variety of applications, including combustion simulations (Bremer *et al.* 2010; Mascarenhas *et al.* 2011), fluid dynamics (Laney *et al.* 2006), climate simulations (Gerber *et al.* 2010), protein folding (Weber *et al.* 2007) and molecular shape analysis (Cazals *et al.* 2003).

This paper describes the R (R Development Core Team 2012) package **msr** (Gerber *et al.* 2012a) for data analysis with the Morse-Smale complex, available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=msr`. The Morse-Smale complex provides a topologically *meaningful* decomposition of the domain of a scalar function. The basic functionality of this package rests on the computation of the Morse-Smale complex from unorganized scattered data. The domain decomposition induced by the Morse-Smale complex is used to build the exploratory visualization technique described in (Gerber *et al.* 2010) and the regression approach in (Gerber *et al.* 2012b).

Gerber *et al.* (2010) exploit the Morse-Smale complex decomposition to form abstract two-dimensional representations of high dimensional scalar functions. These representations present high-level summaries of the salient features of the functions and form the scaffolding for an exploratory data analysis tool. In regression, domain decompositions have been used successfully to build flexible non-parametric models such as regression trees (Breiman *et al.* 1984), multivariate adaptive regression splines (MARS, Friedman 1991) and variations on those approaches (Chaudhuri *et al.* 1994; Alexander and Grimshaw 1996). These partition-based regression methods split the domain recursively into smaller partitions with low-order parametric models in each partition. The recursive partitioning is based on *splitting rules* that typically measure the quality of the resulting fit. In a similar fashion, Gerber *et al.* (2012b) use the domain of the Morse-Smale complex as the splitting rule for a partition-based regression scheme.

To take full advantage of the package presented in this paper, a high-level understanding of the properties of the Morse-Smale complex is necessary. Thus, a brief overview of the Morse-Smale complex, and some issues involving its discrete approximation, is included in this paper. For completeness, the visualization and regression techniques in Gerber *et al.* (2010) and Gerber *et al.* (2012b) are summarized.

2. The Morse-Smale complex

In Morse theory, the Morse-Smale complex provides a tool to examine the topology of a manifold $\mathcal{M}$ based on the critical points of a function f defined on $\mathcal{M}$. Here, the interest is not on the topology of $\mathcal{M}$ but on the exploitation of the Morse-Smale complex to glean information about the geometry of the function f itself.

The Morse-Smale complex partitions the domain of a function f based on the critical points of f. Informally, the interior of each partition is a *monotonic* region, i.e., contains no critical points, as illustrated in Figure 1.

2.1. Formal definition

Let $\mathcal{M}$ be a smooth, compact, p-dimensional manifold. A smooth function $f : \mathcal{M} \mapsto \mathbb{R}$ is *Morse* if, for all critical points x of f, the Hessian matrix $Hf(x)$ is not singular. An *integral line*, $\lambda : \mathbb{R} \mapsto \mathcal{M}$, is a curve in $\mathcal{M}$ with $\frac{d\lambda}{ds}(s) = \nabla f(\lambda(s))$, with $\alpha(\lambda) = \lim_{s\to-\infty} \lambda(s)$ and $\omega(\lambda) = \lim_{s\to\infty} \lambda(s)$. Note, $\alpha(\lambda)$ and $\omega(\lambda)$ are both, by definition, critical points of f. Define the ascending and descending, also referred to as stable and unstable, manifolds of a critical

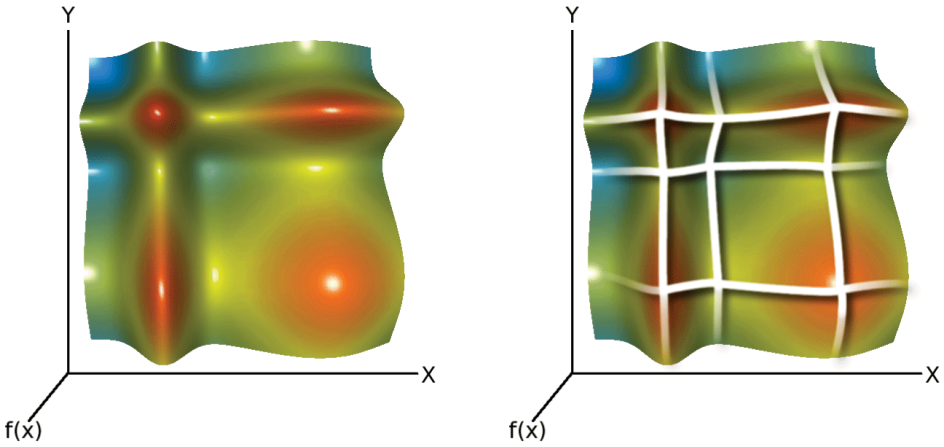

Figure 1: A two-dimensional scalar function (left) and the corresponding Morse-Smale complex (right). The Morse-Smale complex induces a segmentation of the function into monotonic regions using gradient flows with a single local minimum and maximum.

point x as $A(x) = \{\lambda : \alpha(\lambda) = x\}$ and $D(x) = \{\lambda : \omega(\lambda) = x\}$. A Morse function f is Morse-Smale if the ascending and descending manifolds only intersect transversely. The Morse-Smale complex of a Morse-Smale function is the set of intersections, $A(x_i) \cap D(x_j)$, over all combinations of critical points (x_i, x_j). The Morse-Smale complex includes regions (i.e., sub-manifolds of $\mathcal{M}$) of dimensions 0 through p and partitions $\mathcal{M}$. The 0 through $p-1$ dimensional components of the Morse-Smale complex delineate the boundaries of the p-dimensional partitions. By definition, the p-dimensional partitions contain no critical points and have a single local minimum and maximum on the boundary.

2.2. Persistence

The Morse-Smale complex introduces a measure of the significance of each extremal point, called *persistence*. Persistence is a measure of the amount of change in the function f required to *remove* a critical point, and thus, to merge two or more partitions. Note, persistence describes the significance of an extremal point in geometric terms and not in the statistical sense of a hypothesis test. Figure 2 illustrates the concept of persistence simplification on a one-dimensional function.

Definition Let x_i be the critical points of f. Define $s(x_i)$ as the set of critical points that have a direct integral line connecting to x_i. Let $n(x_i) = \arg\min_{x_j \in s(x_i)}(\|f(x_i) - f(x_j)\|$, the persistence of a critical point x_i is then defined as $p(x_i) = \|f(x_i) - f(n(x_i))\|$. This definition is motivated by the amount of change of f in L_∞ required such that the critical point pair $(x_i, n(x_i))$ is either canceled or merged into a single critical point. Recursively removing the critical point with minimal persistence leads to a nested series of successively simplified Morse-Smale complicies, also called a filtration (Edelsbrunner *et al.* 2006; Chazal *et al.* 2009). At each level, some of the partitions induced by the Morse-Smale complex are merged into a single partition until the Morse-Smale partitioning consists of only a single partition (i.e., the entire input domain).

The sequential simplification by persistence leads to a hierarchy of Morse-Smale complicies.

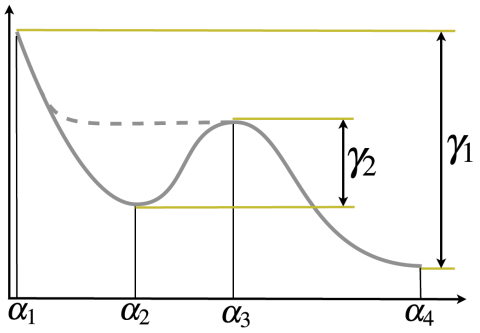

Figure 2: At the highest persistence level, the Morse-Smale complex of the solid gray function has three 1-dimensional partitions (α_1, α_2), (α_2, α_3) and (α_3, α_4). The dashed gray line indicates the change required to remove the critical point pair $\{\alpha_2, \alpha_3\}$ with persistence γ_2, and results in a Morse-Smale complex with a single partition (α_1, α_4).

At the highest persistence level, the Morse-Smale complex only consists of the highest maximum and lowest minimum, and the segmentation corresponds to the entire domain. With decreasing persistence levels, more extremal points and corresponding partitions are introduced in order of their persistence level. Thus, persistence introduces a notion of scale at which the Morse-Smale complex of f is considered.

2.3. Morse-Smale complex approximation

The definition of the Morse-Smale complex, in terms of ascending and descending manifolds, leads to a direct algorithm. Given a set of observations $O = \{y_i, x_i\}_1^N$ with $y_i = f(x_i)$, determine, for each data point x_i, its α- and ω-limit and corresponding partition by following the gradient at x_i. A direct approach would require the estimation of the gradient at x_i, and to trace the integral line that passes through it. This would require essentially an estimate of f. However, the only information required to partition the domain is the α- and ω-limit of the integral line, which is computed using a discrete approximation of the integral line without having to first estimate f.

Note, that the 0 through $p-1$ dimensional components have measure zero. Thus, in practice, the points of a data set are, with probability one, part of the p-dimensional components of the Morse-Smale complex with integral lines ending in local minima and maxima.

In Gerber *et al.* (2010), the domain is approximated via a k nearest-neighbor graph. The algorithm follows paths of steepest ascent and descent based on the connectivity of the graph; essentially a variant of the quick shift algorithm (Vedaldi and Soatto 2008). For such an algorithm, let the adjacencies of point x_i be $\text{adj}(x_i) = \{x_j : x_i \in \text{knn}(x_j), x_j \in \text{knn}(x_i)\}$. Then the approximate integral line is traced out by following the path of steepest ascent $p_a(x_i) = \arg\max_{x_j \in \text{adj}(x_i)} f(x_j) - f(x_i)$ and descent $p_d(x_i) = \arg\max_{x_j \in \text{adj}(x_i)} f(x_i) - f(x_j)$, with $p_a(x_i) = x_i$ and $p_d(x_i) = x_i$ if all neighbors have lower/higher function value (i.e., a local maxima/minima). This assigns each point to a p-dimensional component, identified by local minima/maxima association, of the Morse-Smale complex. Hence, the Morse-Smale approximation results in a partition $C = \{C_1, \ldots, C_n\}$, with C_i the set of points for partition i, such that $\bigcup_i C_i = \{x_i\}_1^n$ and $C_j \cap C_i = \emptyset \forall i \neq j$.

For computing the persistence-based hierarchy of the Morse-Smale complex, an approximation of saddle point values between neighboring partitions is required. Let $s(p_a, p_b)$ be the set of edges (x_i, x_j) such that x_i is assigned to partition p_a and x_j to p_b. Then, the persistence of the minima $a_{\min}$ of partition p_a is defined as $\min_{k \in \bar{P}(X)} \min_{e \in s(p_a, p_k)} \max_{x_i \in e} \|a_{\min} - x_i\|$. Again, as in the continuous case, this is recursively applied with $\bar{P}(X)$, the minimal persistence extrema, removed.

Depending on the number of nearest neighbors and sampling density, *artificial* extremal points can be introduced or *true* extremal points can be removed. Artificial extremal points are introduced if the k-nearest neighbors of a point connect only to points with lower function values. True maxima (or minima) can be removed if the k-nearest neighbors graph introduce a monotonous steepest ascent (or descent) path – strictly positive or negative difference between neighboring points along the path – to another maxima (minima) point. This arises mostly in the case of a large number of nearest neighbors, and/or a sparse sample of the domain.

2.4. Partition prediction

The Morse-Smale complex approximation provides the partition assignments, or labels, for the input data, but not the complete domain. In Gerber *et al.* (2012b), two approaches to extend this partitioning to the complete domain are considered.

The first approach estimates the probability distribution $P(X|C = C_i)$ with a kernel density estimator over the observations s in partition i:

$$p_i(x) = \frac{1}{\|C_i\|} \sum_{s \in C_i} K(x, s)\,. \tag{1}$$

From this, Bayes' theorem yields partition probabilities $P(C = C_i|X = x) = \frac{p_i(x)P(C=C_i)}{\sum_j p_j(x)P(C=C_j)}$ with $P(C = C_i) = \frac{\|C_i\|}{\sum_j \|C_j\|}$.

The second approach employs a support vector machine (SVM) to build a multi-class classifier using the one-against-one approach described in Hsu and Lin (2002). The SVM is trained on the partition assignments of the Morse-Smale complex. To estimate partition probabilities from the classifier, a logistic regression is fit to the decision values of the SVM as described in Platt (1999).

3. The Morse-Smale complex for scalar function visualization

In Gerber *et al.* (2010) the Morse-Smale complex is used to build a summary representation of the high-dimensional scalar function $f : \Omega \mapsto \mathbb{R}$, with Ω a compact connected subset of $\mathbf{R}^n$. Each partition C_i of the Morse-Smale complex is summarized by an inverse regression curve, $r_i(y) = E[X \in C_i|Y = y]$. The regression estimates approximate a graph of curves connecting the minima and maxima as identified by the Morse-Smale segmentation; essentially a compressed representation of the domain. For visualization, the extremal points are projected, using principal components, into two-dimensions (2D). Then, each regression curve is separately projected into 2D, again using principal components. Finally, the projected curves are affine-transformed such that the end points match the corresponding projected extremal points. This concept is illustrated in Figure 3.

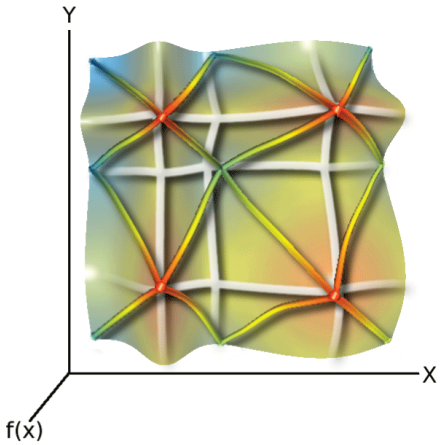

Figure 3: Schematic illustration of the visualization: the two-dimensional scalar function is decomposed into piecewise monotonic partitions by the Morse-Smale complex. The function is then represented by a network of regression curves. Each curve summarizes the domain of a partition with $r_i = E[X \in C_i | y = y]$. For visualization of higher dimensional functions, this network is embedded into 2D.

The visualization created by this method can help to gain a qualitative understanding of higher dimensional functions. In Gerber *et al.* (2010) this exploratory analysis approach is demonstrated on

- climate simulations outcomes,
- chemical compositions in combustion simulations,
- concrete mixture strengths,
- socio-economic variables in relation to crime rates.

The approach consists of three steps to arrive at a 2D representation for visualization of the high-dimensional scalar function:

1. **Morse-Smale approximation:** Compute a segmentation using the Morse-Smale approximation of f.
2. **Geometric summaries:** Construct regression curves $\mathbf{r}_i$ with kernel regression.
3. **Dimension reduction:** Embed the regression curves in 2D using a three-step dimension-reduction approach:
   (a) Project extremal points onto its first two principal components.
   (b) Project each regression curve onto its first two principal components.
   (c) Affine-transform projected curves to match the corresponding projected extremal points.

4. Morse-Smale regression

The idea presented in Gerber *et al.* (2012b) is to use the Morse-Smale complex induced segmentation of the domain of a function $f : \Omega \mapsto \mathbb{R}$ to combine simple linear models within each partition to approximate the underlying function f. Two approaches are considered. First, a piecewise linear model

$$\hat{f}_l(x) = \quad a_j + b_j x \quad \text{if } x \in C_j, \text{ j=1, } \ldots, \text{ k}\,. \tag{2}$$

Second, a sum of weighted linear models, with weights based on the probabilistic interpretation of the segmentation induced by the Morse-Smale complex, i.e., a soft assignment of each point based on the partition probabilities.

$$\hat{f}_w(x) = \sum_{j=1}^{k} w_j(x)(a_j + b_j x) \tag{3}$$

with $w_j(x) = P(C = C_i | X = x)$ estimated as described in Section 2.4.

5. Package description

The methods for approximating the Morse-Smale complex from a sample of a scalar function form the basis of the **msr** package (Gerber *et al.* 2012a). The regression and visualization approaches are built on this basic functionality. This section gives a brief overview of the package. Detailed descriptions of the available methods are in the manual pages contained in the package.

5.1. Morse-Smale complex approximation

The main functions for the Morse-Smale approximation is `msc.nn`. The functions `msc.nn.kd` and `msc.nn.svm` add predictive capacity to `msc.nn` by estimating the partition probabilities with kernel density estimation or a multi-class SVM, respectively (see Section 2.4). All these methods return a basic Morse-Smale complex object `msc.nn` (potentially supplemented with predictive capabilities), which is the basic input for the regression and visualization methods.

The nearest neighbor graph is computed using the approximate nearest neighbor library **ANN** (Mount and Arya 2010). For performance, the kernel density estimation is implemented in C with an interface to R. For the SVM multi-class classifier the R package **e1071** (Dimitriadou *et al.* 2011) is used, which provides an interface to the C++ library **libsvm** (Chang and Lin 2012).

For both `msc.nn.kd` and `msc.nn.svm`, the generic R method `predict` is implemented and returns a matrix with partition probabilities for each point, either of the original data x or, if the argument `newdata` is specified, for the locations of the new observations.

Two additional utility functions are supplied:

- `msc.level.ind` extracts the indices into the original, supplied data x for a given partition index, Morse-Smale complex, and persistence level.
- `msc.sublevels` extracts a subset of the hierarchy of Morse-Smale complicies.

5.2. Visualization

The visualization functionality is accessed through `plot.msc`. This function takes in an `msc.nn` object, computes the necessary geometry for the plots, and creates the visualization. The package **rgl** Adler and Murdoch (2012) is used for three-dimensional display, and user interaction through mouse clicks bring up detailed 2D plots of the inverse regression curves. The plotting function returns an object that allows the user to modify what is displayed in the **rgl** device, such as the optional display of the standard deviation tubes, or the 3D axis.

The visualization displays the graph of the regression curves in 3D. The regression curves are projected into 2D and the 3rd dimension is used to encode the function value, also redundantly encoded with a colormap. While color provides a quick overview of the location of minima and maxima, the 3rd dimension can be used to better perceive quantitative differences in minima and maxima and helps to visualize the function as a height field over the graph representation of the domain.

Opaque tubes surrounding the curves encode the standard deviation of the regression curves by mapping to the radius of the tubes. The standard deviation provides information about the shape of a partition, i.e., the *extent* of the level sets.

Figure 4 shows the use of the 3rd dimension as well as the standard deviation tubes, which can be turned on or off through a user parameter. In addition, the user can inspect the behavior of the independent variables in each partition by clicking on the corresponding regression curve which then highlights that curve in an identifying color in a corresponding 2D graph, as shown in Figure 5, left. A separate graphics device is then opened, and plots the regression curve $r_i(y) = E[X \in C_i | Y = y$ with d graphs of y as a function of $r_i(y)$ for each dependent variable in X (Figure 5, right). To compare various partitions simultaneously, the user can

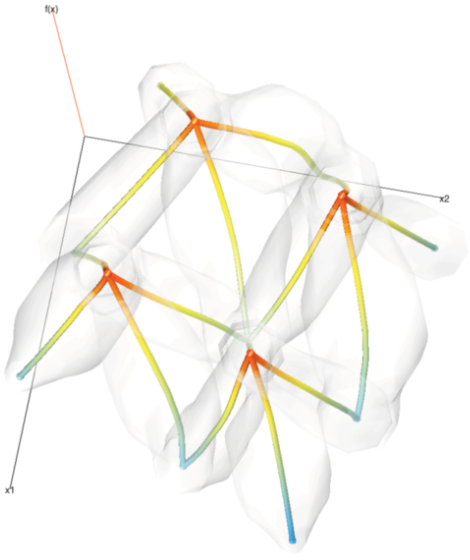

Figure 4: Visual summary representation from 2000 observations of the function in Figure 1. The regression curves for each partition are projected into 2D with the 3rd dimension encoding the function value. The function value is redundantly encoded using a colormap. The standard deviation of the regression curve is displayed by the opaque tube surrounding the curve.

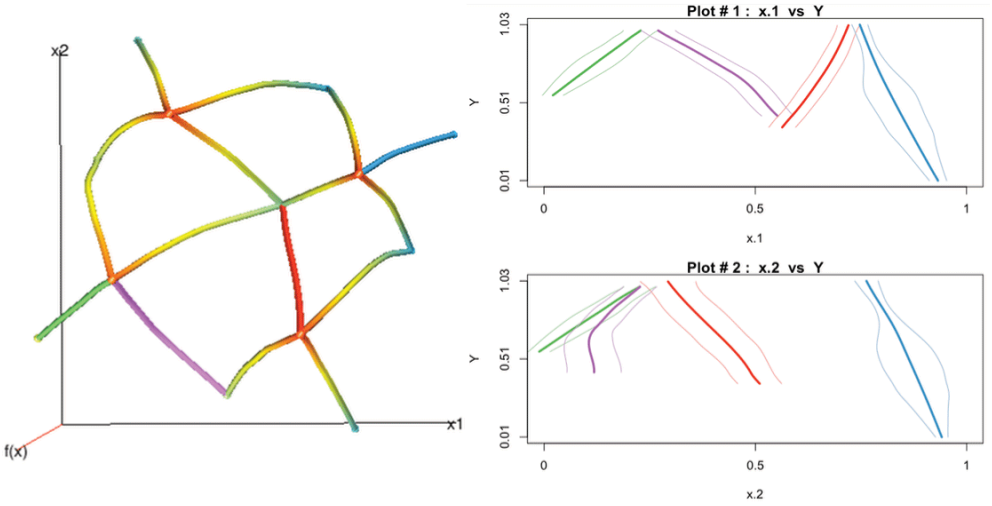

Figure 5: The interactive visualization of the data set from Figure 1. The user is presented with the visual summary (left) and can select curves of interest (highlighted in red, green, purple and blue). These curves, representing independent variables, are then plotted side-by-side (right).

select multiple curves, which are displayed in the same plot using the matching color from the main display.

5.3. Regression

The piecewise linear model for the Morse-Smale regression is implemented in the two methods:

- `msc.lm`: The piecewise linear model.
- `msc.elnet`: The piecewise linear model with L_1 regularization using the package **glmnet** (Friedman *et al.* 2010).

and for the weighted linear model:

- `msc.slm`: The weighted linear model.
- `msc.slm.elnet`: The weighted linear model with L_1 regularization using **glmnet**.

The methods take as input a Morse-Smale complex object and additional parameters that influence the fitting.

6. A complete tour

This section gives a complete tour of the package on a simple 2D function, for which it is easy to show visual results for all aspects of the package. The tour is based on the 2D function shown in Figure 1, which is defined by

$$f(x) = \frac{1}{2}\left(e^{-(x_1-\frac{1}{4})^2/0.3^2} + e^{-(x_2-\frac{1}{4})^2/0.3^2} + e^{-(x_1-\frac{3}{4})^2/0.1^2} + e^{-(x_2-\frac{3}{4})^2/0.1^2}\right). \quad (4)$$

6.1. Morse-Smale complex construction and partition prediction

Create 2000 observations from f, uniformly distributed on $[0,1]^2$ (scaled to $[0,1]$ for color-mapping).

```
R> library("msr")
R> data("fourpeaks")
R> d <- fourpeaks(2000)
R> d[, 1] <- (d[, 1] - min(d[, 1])) / (max(d[, 1]) - min(d[, 1]))
```

and compute the Morse-Smale complex at persistence level 0.1 with 15 nearest neighbors.

```
R> ms <- msc.nn(y = d[, 1], x = d[, 2:3], pLevel = 0.1, knn = 15)
R> summary(ms)
```

Plot the function and the segmentation induced by the Morse-Smale complex in 2D (see Figure 6).

```
R> par(mfcol = c(1, 2))
R> fancy <- c("#0066CC", "#CCCC00", "#D22905")
R> colormap <- colorRamp(fancy, interpolate = "linear", bias = 0.5)
R> colors <- rgb(colormap(d[, 1]), maxColorValue = 255)
R> par(mar = c(8, 5, 5, 4))
R> plot(d[, 2], d[, 3], type = "p", xlab = "", ylab = "", col = colors,
+    pch = 19, cex.axis = 2.5)
R> pal <- brewer.pal(9, "Set1")
R> pal <- colorRampPalette(pal)
R> pal <- pal(length(ms$level[[1]]$mins))
R> colors <- pal[ms$level[[1]]$partition]
R> par(mar = c(8, 5, 5, 4))
R> plot(d[, 2], d[, 3], col = colors, type = "p", xlab = "", ylab = "",
+    pch = 19, cex.axis = 2.5)
```

Show the number of extrema as a function of persistence (see Figure 7).

```
R> np <- length(ms$persistence)
R> ms$persistence[np] <- 1
R> par(mar = c(6, 4, 4, 4) + 3)
R> plot(ms$persistence, np:1 + 1, xlab = "Persistence Percentage",
+    ylab = "Extrema", cex = 2, cex.lab = 2, cex.axis = 2, type = "p",
+    pch = 19)
```

The persistence is expressed as a percentage of the function range. Note, that `persistence` always contains the complete persistence hierarchy, even if only a subset of the complete Morse-Smale complex persistence hierarchy is computed.

In the above example, the Morse-Smale complex was computed for a fixed persistence level. Alternatively, compute the last 15 persistence levels of the Morse-Smale complex

```
R> ms <- msc.nn(y = d[, 1], x = d[, 2:3], nLevels = 15, knn = 15)
```

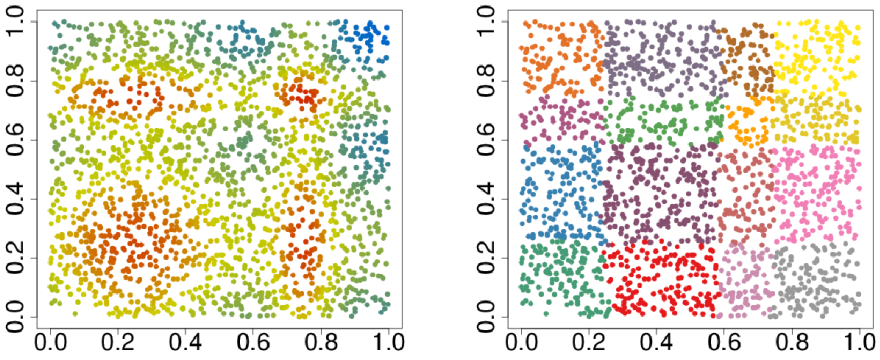

Figure 6: Function and segmentation induced by Morse-Smale complex in 2D.

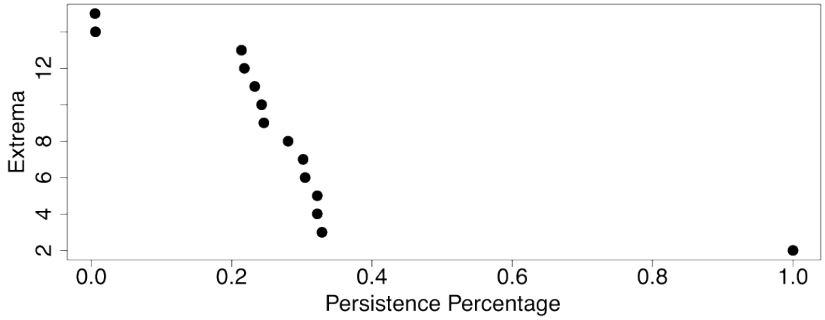

Figure 7: Number of extrema as a function of persistence.

and access individual levels.

```
R> level5 <- ms$level[[5]]
R> summary(level5)
```

```
              Length Class  Mode
call             1   -none- function
pLevel           1   -none- numeric
partition     2000   -none- numeric
maxs             6   -none- numeric
mins             6   -none- numeric
partitionSize    6   -none- numeric
```

Note that the persistence hierarchy is in reverse order, starting from highest persistence to `nLevels` lower persistences Morse-Smale complicies.

At this point, the Morse-Smale complex has no prediction capabilities. Compute a Morse-Smale complex that supports partition prediction by SVM.

```
R> ms.svm <- msc.nn.svm(y = d[, 1], x = d[, 2:3], nLevels = 15, knn = 15,
+    cost = 1, precompute = TRUE)
```

The `precompute` flag indicates that the SVM at each persistence level is computed and stored for later use. If set to `FALSE`, the SVM is computed on demand, which saves storage space, but can be time consuming if multiple predictions at different levels are made.

Alternatively, construct a Morse-Smale complex with partition prediction by kernel density estimation for two different bandwidths.

```
R> ms.kd1 <- msc.nn.kd(y = d[, 1], x = d[, 2:3], nLevels = 15, knn = 15,
+    bw = 0.05)
R> ms.kd2 <- msc.nn.kd(y = d[, 1], x = d[, 2:3], nLevels = 15, knn = 15,
+    bw = 0.15)
```

Create a test data set and predicts partition probabilities at level 12.

```
R> test <- fourpeaks(2000)
R> ms.svm$predictLevel <- 12
R> psvm <- predict(ms.svm, newdata = test)
R> ms.kd1$predictLevel <- 12
R> ms.kd2$predictLevel <- 12
R> pkd1 <- predict(ms.kd1, newdata = test)
R> pkd2 <- predict(ms.kd2, newdata = test)
```

For both visualization and regression, `predictLevel` determines the level of the current Morse-Smale hierarchy at which prediction or visualization is performed, with `predictLevel = 1` corresponding to the highest peristence.

Now plot the partition probabilities for partition index 8 (see Figure 8).

```
R> par(mfcol = c(1, 4))
R> pId <- ms.svm$level[[12]]$partition == 8
R> colors <- vector(length = length(ms.svm$y))
R> colors[] <- fancy[1]
R> colors[pId] <- fancy[3]
R> par(mar=c(6, 4, 4, 4) + 1)
R> plot(ms$x, col = colors, type = "p", xlab = "", ylab = "", pch = 19,
+    cex.axis = 2, cex = 1)
R> title("Ground Truth", cex.main = 3.5)
R> colors <- rgb(colormap(psvm[, 8]), maxColorValue = 255)
R> par(mar=c(6,4,4,4) + 1)
R> plot(test[, 2], test[, 3], type = "p", xlab = "", ylab = "",
+    col = colors, pch = 19, cex.axis = 2, cex = 1)
R> title("SVM", cex.main = 3.5)
R> colors <- rgb(colormap(pkd1[, 8]), maxColorValue = 255)
```

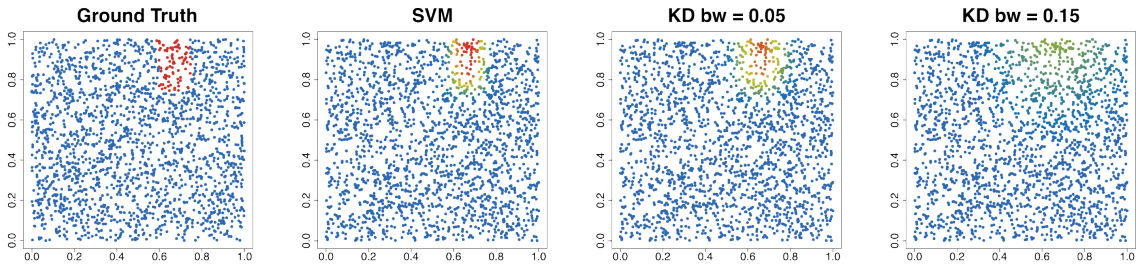

Figure 8: Partition probabilities.

```
R> par(mar = c(6, 4, 4, 4) + 1)
R> plot(test[, 2], test[, 3], type = "p", xlab = "", ylab = "",
+    col = colors,  pch = 19, cex.axis = 2, cex = 1)
R> title("KD bw = 0.05", cex.main = 3.5)
R> colors <- rgb(colormap(pkd2[, 8]), maxColorValue = 255)
R> par(mar = c(6, 4, 4, 4) + 1)
R> plot(test[, 2], test[, 3], type = "p", xlab = "", ylab = "",
+    col = colors, pch = 19, cex.axis = 2, cex = 1)
R> title("KD bw = 0.15", cex.main = 3.5)
```

6.2. Visualization

Using the Morse-Smale complex from Section 6.1 show visualizations at level 2, 7, and 12 (see Figure 9).

```
R> ms$predictLevel <- 2
R> plot(ms, span = 0.9)
R> ms$predictLevel <- 7
R> plot(ms)
R> ms$predictLevel <- 12
R> plot(ms)
```

As for the partition prediction, the `predictLevel` defines at which persistence level the visualization is performed.

6.3. Regression

The regression is invoked directly on the Morse-Smale complex object. Construct Morse-Smale complex with SVM prediction and fit a piecewise linear model with a cross-validation for choosing the prediction level.

```
R> ms.svm$predictLevel <- 1
R> msr.lm <- msc.lm(ms.svm)
R> summary(msr.lm)

      Length Class   Mode
lms   15     -none-  list
```

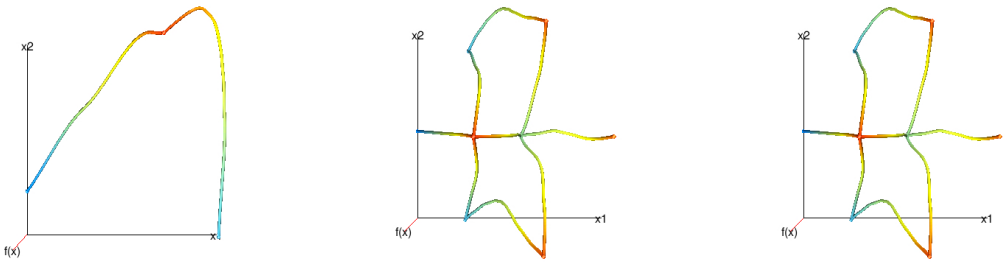

Figure 9: Morse-Smale complex visualizations at level 2, 7, and 12.

```
ms     9     msc.svm list
blend  1     -none-  logical
```

```
R> pLevel <- msr.lm$ms$predictLevel
R> print(pLevel)
```

```
[1] 14
```

```
R> print(msr.lm$lms[[pLevel]]$cv)
```

```
           1
0.001819926
```

This constructs a hierarchy of linear models and the `msr$ms$predictLevel` designates the model used for prediction. Predict the test data and print the mean squared error.

```
R> p <- predict(msr.lm, newdata = test)
R> print(mean((p - test[, 1])^2))
```

```
[1] 0.003417632
```

Instead of the piecewise linear model, fit a sum of weighted linear models. The persistence level is, as for the piecewise linear model, selected automatically with cross-validation.

```
R> msr.slm <- msc.slm(ms.svm)
R> print(msr.slm$slm[[pLevel]]$cv)
```

```
[1] 0.00140596
```

Predict at different levels and plot the corresponding models (see Figure 10).

```
R> msr.slm$ms$predictLevel <- 1
R> p1 <- predict(msr.slm, newdata = test)
R> print(mean((p1 - test[, 1])^2))
```

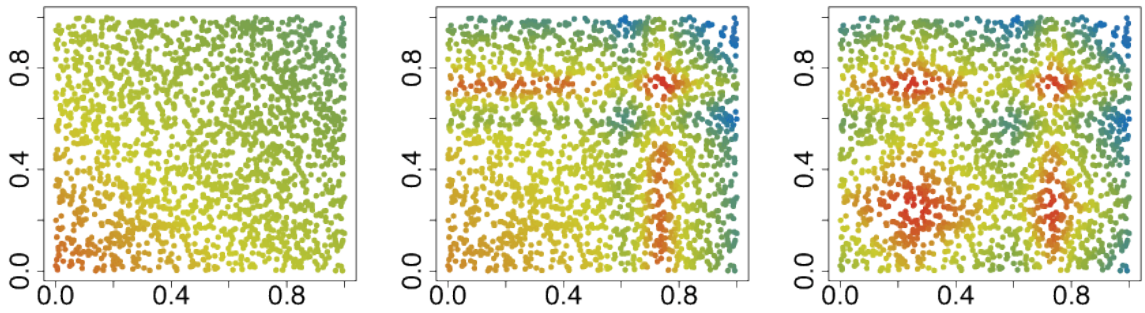

Figure 10: Model visualization at different levels.

```
[1] 0.03254943

R> msr.slm$ms$predictLevel <- 7
R> p2 <- predict(msr.slm, newdata = test)
R> print(mean((p2 - test[, 1])^2))

[1] 0.01055478

R> msr.slm$ms$predictLevel = 12
R> p3 <- predict(msr.slm, newdata = test)
R> print(mean((p3 - test[, 1])^2))

[1] 0.002921932

R> par(mfcol = c(1, 3))
R> ps <- p1
R> ps[ps > 1] <- 1
R> ps[ps < 0] <- 0
R> colors <- rgb(colormap(ps), maxColorValue = 255)
R> plot(test[,2], test[,3], type = "p", xlab = "", ylab = "", col = colors,
+    pch = 19, cex.axis = 2.5, cex = 1)
R> ps <- p2
R> ps[ps > 1] <- 1
R> ps[ps < 0] <- 0
R> colors <- rgb(colormap(ps), maxColorValue = 255)
R> plot(test[,2], test[,3], type = "p", xlab = "", ylab = "", col = colors,
+    pch = 19, cex.axis = 2.5, cex = 1)
R> ps <- p3
R> ps[ps > 1] <- 1
R> ps[ps < 0] <- 0
R> colors <- rgb(colormap(ps), maxColorValue=255)
R> plot(test[,2], test[,3], type = "p", xlab = "", ylab = "", col = colors,
+    pch = 19, cex.axis = 2.5, cex = 1)
```

7. Visualizing the UCI crime data

The communities and crime data set from the UCI machine learning repository (Frank and Asuncion 2010) combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. The original data set contains 1,994 observations (communities) with 128 attributes. This analysis uses a reduced data set, with attributes that have missing entries removed, and investigates the crime rate in relation to 99 socio-economic variables.

```
R> data("uci_crime_subset")
```

First, construct the Morse-Smale complex and plot the persistences.

```
R> ms <- msc.nn(y = crimes[, 100], x = crimes[, 1:99], knn = 120,
+    nLevels = 5)
R> np <- length(ms$persistence)
R> ms$persistence[np] <- 1
R> par(mar = c(5, 4, 4, 2) + 3)
R> plot(ms$persistence, np:1 + 1, xlab = "Persistence Percentage",
+    ylab = "Extrema", cex = 2.5, cex.lab = 2, cex.axis = 2, type = "p",
+    pch = 19)
```

This shows that the Morse-Smale complex with 4 extrema requires a function change of 26 percent of the function range to introduce an additional extrema (see Figure 11). This indicates that the Morse-Smale complex at this level is relatively stable and the extremal points are not very likely artifacts from noisy observations.

Now plot the Morse-Smale based summary representation of the function at persistence level 3 (see Figure 12).

```
R> ms$predictLevel <- 3
R> obj <- plot(ms)
```

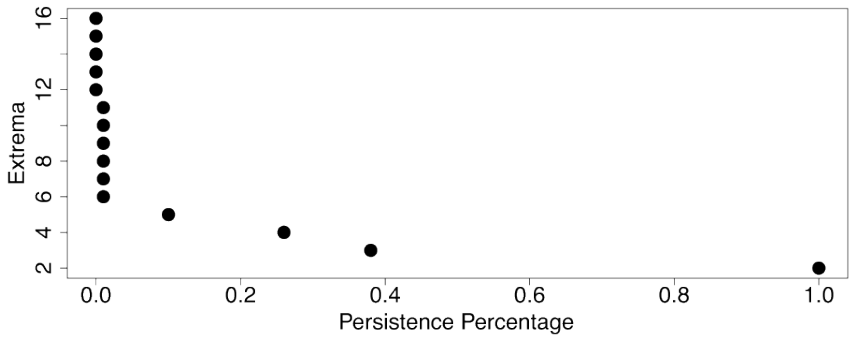

Figure 11: Number of extrema as a function of persistence.

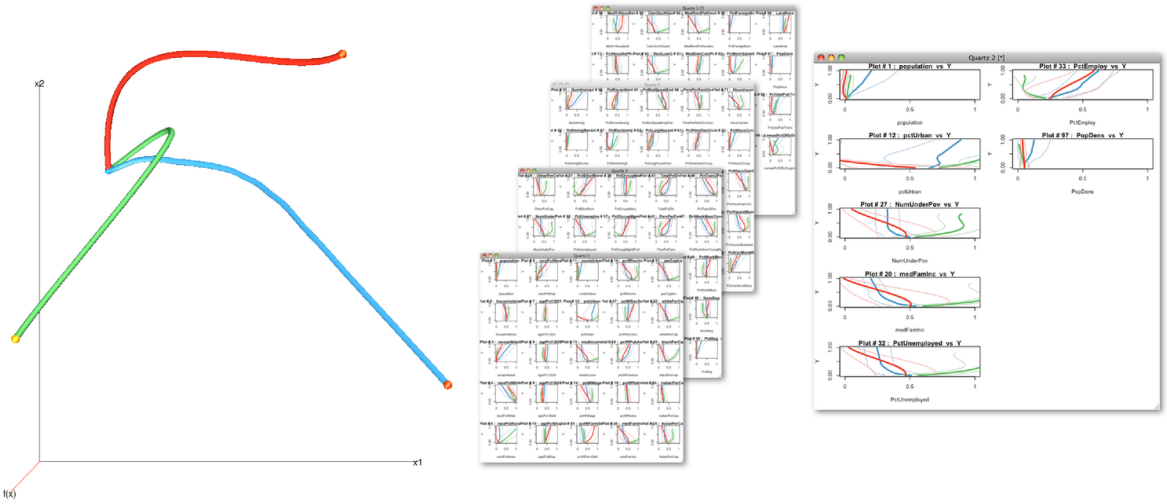

Figure 12: Visual summary of the UCI crime data set with the regression curves for all 99 variables and a selected subset of 7 variables.

This visualization is interactive and allows for deeper investigation of the function. By clicking on a regression curve, representing one partition of the Morse-Smale complex, a window with the inverse regression curves per variable is shown for that curve. In the case of the crime data, this opens multiple device windows to show the 99 variables of the domain. This can be cumbersome to analyze. To restrict attention to variables of interest, a `plotList` on the object is created which allows the user to select specific variables for plotting.

```
R> obj$plotList <- c(1, 12, 27, 20, 32, 33, 97)
R> names <- colnames(crimes)[obj$plotList]
R> print(names)
R> plot(obj)
```

This allows for the easy comparison of user selected variables of interest. For example, the selected variables show that two peak crime rate configurations involve a large percentage of urban population, while the third high crime rate peak concerns rural areas, One peak crime rate is related to high unemployment rates while the other two have relatively low unemployment. This illustrates how the proposed visualization can be used to build hypotheses for further examination.

8. A Morse-Smale regression example

This example is based on the energy/objective function of a camera estimation problem.

```
R> data("camera_estimation")
R> energy$E <- (energy$E - min(energy$E)) / (max(energy$E) - min(energy$E))
R> train <- energy[1:2000, ]
R> test <- energy[8001:10000, ]
```

Create a Morse-Smale complex object with kernel density based partition prediction.

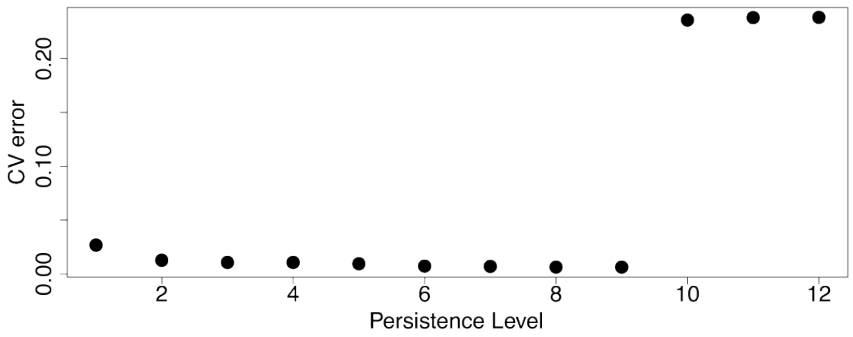

Figure 13: Cross-validation error as a function of the number of extrema.

```
R> ms <- msc.nn.kd(y = train$E, x = train[, 2:10], knn = 40, nLevels = 17,
+    bw = 0.3)
```

The piecewise linear model

```
R> ms.lm <- msc.lm(ms)
```

at level

```
R> pLevel <- ms.lm$ms$predictLevel
R> print(pLevel)
```

```
[1] 9
```

has the lowest cross-validation error. Plot the cross-validation error as a function of the number of extrema (see Figure 13).

```
R> cv <- c()
R> for(i in 1:ms.lm$ms$nLevels) cv[i] <- ms.lm$lms[[i]]$cv
R> par(mar = c(5, 4, 4, 2) + 3)
R> plot(1:ms.lm$ms$nLevels, cv, xlab = "Persistence Level",
+    ylab = "CV error", pch = 19, cex = 2.5, cex.lab = 2, cex.axis = 2)
```

The segmentation from the Morse-Smale complex can be used to compute an ANOVA with the partition id as factor.

```
R> df <- data.frame(test, pID =  as.factor(ms$level[[pLevel]]$partition))
R> lm1 <- lm(E ~ . * pID, data = df)
```

Here, the different partitions are, except for a few (visible with `summary(lm1)`, not shown), not significantly different. However, in general, this illustrates how the partition based regression

can be used to detect different regions of the parameter space where the function shows different behaviours.

Next, extract a single persistence level Morse-Smale complex and estimate a piecewise linear model with an L_1 regularization

```
R> ms15 <- msc.sublevels(ms, pLevel)
R> ms.l1 <- msc.elnet(ms15)
```

and show coefficients for partition 1 and 5

```
R> b1 <- as.vector(ms.l1$elnet[[1]]$lm[[1]]$beta)
R> b5 <- as.vector(ms.l1$elnet[[1]]$lm[[5]]$beta)
R> print(cbind(b1, b5))
```

```
                 b1          b5
 [1,]  0.1384267871  0.06085361
 [2,]  0.0563464331  0.05729387
 [3,]  0.0380388475 -0.13577235
 [4,] -0.0003342101 -0.07801561
 [5,]  0.4440587959  0.44355786
 [6,]  0.0049414605 -0.01630017
 [7,] -0.0433647106  0.11108694
 [8,]  0.1011040420  0.13535760
 [9,]  0.2020158713  0.18104548
```

to compare differences between the partitions.

Alternatively, employ a forward stepwise approach for a sparse regression estimate.

```
R> ms.lm1 <- msc.lm(ms15, modelSelect = TRUE)
```

Next, construct a weighted additive.

```
R> ms.slm <- msc.slm(ms)
```

The cross-validation selected persistence level

```
R> print(ms.slm$ms$predictLevel)
```

```
[1] 12
```

with a mean squared test error of

```
R> penergy <- predict(ms.slm, test)
R> mean((test$E - penergy)^2)
```

```
[1] 0.002840482
```

Note, that due to the averaging of different linear models the additive weighted approach is not suitable for interpretation, but typically yields better function approximations than the piecewise linear approach.

9. Conclusion

This paper described the R package **msr** for exploratory data analysis based on the Morse-Smale complex approximation of a function. The package is built in a modular fashion with the Morse-Smale complex approximation as the fundamental building block. A visualization and a regression approach are implemented on top of the Morse-Smale approximation. The regression and visualization facilitate a qualitative understanding of high-dimensional functions. However, the modular design enables to harness the Morse-Smale complex for data analysis approaches other than the two methods presented here.

Acknowledgments

We thank Oliver Rübel for early tests and many bug reports as well as helpful discussions on the design of the package and Peter G. Lindstrom for providing us with the optimization data set. We thank the anonymous reviewers for their helpful comments and technical clarifications.

This work was funded by the National Institute of Health grants U54-EB005149 and 2-P41-RR12553-08, NSF grant CCF-073222 and CNS-0615194, and Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

References

Adler D, Murdoch D (2012). ***rgl**: 3D Visualization Device System (OpenGL)*. R package version 0.92.879, URL http://CRAN.R-project.org/package=rgl.

Alexander WP, Grimshaw SD (1996). "Treed Regression." *Journal of Computational and Graphical Statistics*, **5**(2), 156–175.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). *Classification and Regression Trees*. Wadsworth, California.

Bremer PT, Weber G, Pascucci V, Day M, Bell J (2010). "Analyzing and Tracking Burning Structures in Lean Premixed Hydrogen Flames." *IEEE Transactions on Visualization and Comper Graphics*, **16**(2), 248–260.

Carr H, Snoeyink J, Axen U (2003). "Computing Contour Trees in All Dimensions." *Computational Geometry: Theory and Applications*, **24**(3), 75–94.

Cazals F, Cazal F, Lewiner T (2003). "Molecular Shape Analysis Based Upon Morse-Smale Complex and the Connolly Function." In *Proceedings of the ACM Symposium on Computational Geometry (SOCG)*, pp. 351–360.

Chang CC, Lin CJ (2012). ***LIBSVM**: A Library for Support Vector Machines*. URL http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chaudhuri P, Huang MC, Loh WY, Yao R (1994). "Piecewise-Polynomial Regression Trees." *Statistica Sinica*, **4**, 143–167.

Chazal F, Guibas LJG, Oudot SY, Skraba P (2009). "Analysis of Scalar Fields over Point Cloud Data." In *SODA '09: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1021–1030.

Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A (2011). ***e1071**: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6, URL http://CRAN.R-project.org/package=e1071.

Edelsbrunner H, Harer J, Natarajan V, Pascucci V (2003). "Morse-Smale Complexes for Piecewise Linear 3-Manifolds." In *Proceedings of the 19th Symposium on Compututational Geometry (SOCG)*, pp. 361–370.

Edelsbrunner H, Morozov D, Pascucci V (2006). "Persistence-Sensitive Simplification of Functions on 2-Manifolds." In *Proceedings of the ACM Symposium on Computational Geometry (SOCG)*, pp. 127–134.

Frank A, Asuncion A (2010). "UCI Machine Learning Repository." University of California, Irvine, School of Information and Computer Sciences, URL http://archive.ics.uci.edu/ml/.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software*, **33**(1), 1–22. URL http://www.jstatsoft.org/v33/i01/.

Friedman JH (1991). "Multivariate Adaptive Regression Splines." *The Annals of Statistics*, **19**(1), 1–141.

Gerber S, Bremer PT, Pascucci V, Whitaker RT (2010). "Visual Exploration of High Dimensional Scalar Functions." *IEEE Transactions on Visualization and Computer Graphics*, **16**(6), 1271–1280.

Gerber S, Potter K, Rübel O (2012a). ***msr**: Morse-Smale Approximation, Regression and Visualization*. R package version 0.4.1, URL http://CRAN.R-project.org/package=msr.

Gerber S, Rübel O, Bremer PT, Pascucci V, Whitaker RT (2012b). "Morse-Smale Regression." *Journal of Computational and Graphical Statistics*. doi:10.1080/10618600.2012.657132. Forthcoming.

Gyulassy A, Natarajan V, Pascucci V, Hamann B (2007). "Efficient Computation of Morse-Smale Complexes for Three-Dimensional Scalar Functions." *IEEE Transactions on Visualization and Computer Graphics*, **13**(6), 1440–1447.

Hsu CW, Lin CJ (2002). "A Comparison of Methods for Multiclass Support Vector Machines." *IEEE Transaction on Neural Networks*, **13**(2), 415–425.

Laney D, Bremer PT, Mascarenhas A, Miller P, Pascucci V (2006). "Understanding the Structure of the Turbulent Mixing Layer in Hydrodynamic Instabilities." *IEEE Transactions on Visualization and Computer Graphics*, **12**(5), 1052–1060.

Mascarenhas A, Grout RW, Bremer PT, Hawkes ER, Pascucci V, Chen JH (2011). "Topological Feature Extraction for Comparison of Terascale Combustion Simulation Data." In V Pascucci, X Tricoche, H Hagen, J Tierny (eds.), *Topological Methods in Data Analysis and Visualization*, Mathematics and Visualization, pp. 229–240. Springer-Verlag, Berlin.

Mount DM, Arya S (2010). *Approximate Nearest Neighbors **ANN** Library*. C library version 1.1.2, URL http://www.cs.umd.edu/~mount/ANN/.

Platt JC (1999). "Probabilities for SV Machines." In AJ Smola, PL Bartlett, B Schölkopf, D Schuurmans (eds.), *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press, Cambridge.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Van Kreveld MJ, Van Oostrum R, Bajaj CL, Pascucci V, Schikore D (1997). "Contour Trees and Small Seed Sets for Isosurface Traversal." In *Symposium on Computational Geometry*, pp. 212–220.

Vedaldi A, Soatto S (2008). "Quick Shift and Kernel Methods for Mode Seeking." In *Proceedings of the European Conference on Computer Vision*, pp. 705–718.

Weber G, Bremer PT, Pascucci V (2007). "Topological Landscapes: A Terrain Metaphor for Scientific Data." *IEEE Transactions on Visualization and Computer Graphics*, **13**(6), 1077–2626.

Affiliation:

Samuel Gerber
Scientific Computing and Imaging Institute
University of Utah
72 S Central Campus Drive, Room 3750
Salt Lake City, UT 84112, United States of America
E-mail: sgerber@cs.utah.edu
URL: http://www.cs.utah.edu/~sgerber/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 50, Issue 2 *Submitted:* 2011-03-22
July 2012 *Accepted:* 2012-04-13