s c c h
software competence center
hagenberg

FUZZY.LOGIC.LAB.LINZ

Advances in Knowledge-Based Technologies

Proceedings of the Master and PhD Seminar Summer term 2007, part 1

Softwarepark Hagenberg
SCCH, Room 0/10
May 10, 2007

Software Competence Center Hagenberg
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

14:00–15:30 Session 1 ***(Chair: Bernhard Moser)***

14:00 Julian Mattes:
Biomedical Data Analysis

14:30 Alfredo Lopez:
Motion Analysis of Medical Structures

15:00 Ulrich Bodenhofer:
Towards a Robust Rank Correlation Measure for Numerical Observations on the Basis of Fuzzy Orderings

15:30 Coffee Break

15:45–17:15 Session 2 ***(Chair: Roland Richter)***

15:45 Zheng He:
Learning of Decision Trees with Incremental ID3

16:15 Matej Smid:
Case Study of RANSAC for Image Registration

16:45 Frank Bauer:
Newton Methods for Nonlinear Inverse Problems with Random Noise

17:15 Werner Groissböck:
HDFormGen: A Fast Nonlinear Approximation Formula Generator for Very High Dimensional Data Based on Variable Selection and Genetic Programming

ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Methods 29 (2003) 3–13

METHODS

www.elsevier.com/locate/ymeth

Quantitative motion analysis and visualization of cellular structures

Daniel Gerlich,*,[1] Julian Mattes, and Roland Eils

Intelligent Bioinformatics Systems, DKFZ, Im Neuenheimer Feld 280, Heidelberg D-69120, Germany

Accepted 11 September 2002

Abstract

The availability of cellular markers tagged with the green fluorescent protein (GFP) has recently allowed a large number of cell biological studies to be carried out in live cells, thereby addressing the dynamic organization of cellular structures. Typically, microscopes capable of video recording are used to generate time-resolved data sets. Dynamic imaging data are complex and often difficult to interpret by pure visual inspection. Therefore, specialized image processing methods for object detection, motion estimation, visualization, and quantitation are required. In this review, we discuss concepts for automated analysis of multidimensional image data from live cell microscopy and their application to the dynamics of cell nuclear subcompartments.

Keywords: Four-dimensional imaging; Live cell microscopy; Graphical visualization; Motion analysis; Single-particle tracking; Segmentation; Registration; Optical flow; Active contour; Confinement tree

1. Introduction

All biological phenomena are dynamic. However, for a long time cellular structures have been investigated mostly in fixed specimens, e.g., by the use of immunocytochemistry or fluorescence in situ hybridization, due mainly to the lack of efficient vital markers. This has recently changed with the cloning of green fluorescent protein (GFP) as a universal fluorescent marker that can be fused to many proteins to visualize virtually any cellular structure in the environment of the living cell [1,2].

Live cell studies have revealed the unexpectedly high dynamics of many cellular structures including nuclear subcompartments that were previously thought to be of rather stable morphology (e.g., [3–5]). Early live cell studies have characterized organelle dynamics in a qualitative way. A simple approach to estimate morphological alterations over time or the velocity with which an organelle moves within the cell is to interpret time-lapse movies by visual inspection. Although this can be helpful in obtaining an overall impression of motion patterns, it is not suitable for addressing underlying mechanisms using functional assays carried out in live cells. For example, the dynamics of several nuclear subcompartments have been observed to be regulated within the cell [3,6–10], dependent on cell cycle state or metabolic energy. A convincing and reproducible presentation of functional assays addressing the dynamics of cellular structures under different experimental conditions requires quantitation by automated image processing techniques. In this review, we discuss techniques for quantitative analysis and visualization of live cell imaging data and exemplify their applicability by analysis of the dynamics of several nuclear subcompartments.

2. Computational methods for quantitative analysis and visualization of live cell imaging data

The analysis of live cell imaging data usually comprises the isolation (segmentation) and tracking of fluorescent structures in the time series. This leads to the extraction of space- and time-related data and further interpretation is conducted on those data. For practical

* Corresponding author.
E-mail address: daniel.gerlich@embl.de (D. Gerlich).
[1] Present address: Gene Expression and Cell Biology/Biophysics Programmes, EMBL, Meyerhofstrasse 1, Heidelberg D-69117, Germany.

1046-2023/02/$ - see front matter © 2002 Elsevier Science (USA). All rights reserved.
doi:10.1016/S1046-2023(02)00287-6

applications, these image analysis modules have been combined in entirely different ways. For instance, to characterize the movement of a cell [11] the parameters of an affine transformation (mapping) have been estimated, based directly on the image gray values, thus no segmentation was necessary (see Section 2.2.2). For interpretation, a mathematical analysis of the parameters of the affine transformation has been carried out and the velocity vector field has been visualized. In contrast, other algorithms perform tracking of multiple presegmented objects, taking into account only the gravity center and some attributes of each object [12]. Still other procedures extract the object surface or contour, quantify the surface area and volume based on the extracted data [13], and estimate the transformation that maps the whole image volume such that corresponding surface points in different time steps are assigned to each other, which is called point set registration [14,15]. In the following, we review different elementary image processing techniques according to the sequential order in which they are usually applied. In particular, we discuss a selected set of techniques that are especially suitable for motion analysis of cellular structures and compartments.

2.1. Image segmentation

The first step in image analysis generally is to segment the image. Segmentation subdivides an image into its constituent parts or objects, which are defined as homogenous and disjoint regions (image segments) that are separated by boundaries. Therefore, a criterion of homogeneity has to be found that can be very specific in the different areas of applications. In principle, two opposite approaches can be distinguished. Contour-oriented segmentation uses methods to detect differences between neighboring image points and is therefore based on the image gradients. Contours are drawn between homogenous image segments at locations where the gradient is strong. In contrast, region-oriented segmentation is focused on the detection of similarities of neighboring pixels that are then merged and assigned to image segments. Depending on the initialization of the contour and on the choice of the respective parameters, both approaches should lead to the same segmentation result. However, in practice the performance of both approaches is often very different and strongly depends on the specific application. To unify them a method called ''region competition'' has been proposed [16]. Independent of the segmentation technique images often need to be preprocessed for efficient segmentation.

2.1.1. Image preprocessing by linear convolution filters

Images acquired by live cell microscopy may be disturbed by a variety of noise sources. Noise is often introduced during the conversion of patterns of light energy into electrical patterns in the recording device (e.g., CCD camera or photomulitplier tube). Among the different types of noise that affect an image at different spatial frequencies, the high-frequency noise, which causes loss of sharpness and alters the image contours, often has to be reduced by filtering for further image processing.

Low-pass filtering is employed to remove this high-spatial-frequency noise from a digital image. The basic idea is that a window of some finite size and shape is scanned across the image. The output pixel value is the weighted sum of the input pixels within the window where the weights are the values of the filter assigned to every pixel of the window itself. The window with its weights is called the convolution kernel. A typical shape of the convolution kernel is modeled by a Gaussian function.

2.1.2. Nonlinear filters

Many of the noise-reducing filters are based on low-pass filtering, and are therefore suitable for reducing the spot noise, which is located mostly at high spatial frequencies. Unfortunately, because contours and edges in an image are mainly high frequency, while using low-pass filters to reduce noise, one loses information about important features of an image (blurring effect). This occurs because of the nonspecificity of the filters used that simply cut off high spatial frequencies, no matter what the pixel can represent in an image. In contrast to linear filters, with nonlinear filters any function of the neighborhood pixels can be defined, enabling noise reduction without blurring.

The median filter is the most common nonlinear filter [17]. It replaces every pixel of an image with the median of the pixel intensities in a neighborhood. For a pixel near an edge, pixels on the same side of the edge will be in a majority in a square or circular neighborhood, and so the median will be within the intensity range of pixels on that side. Thus edges remain sharp, unlike those with the linear filters. This method is particularly effective when the noise pattern consists of strong, spike-like components and the characteristic to be preserved is edge sharpness.

2.1.3. Anisotropic diffusion filters

A new generation of more sophisticated filters have been developed, which take into account the particular characteristics of local image features and thus perform a filtering operation without degrading the image. The approach of this type of filter is based on the statistical properties of the noise. The rationale of this method is that image areas containing structure and strong contrast between edges will have a higher variance than areas containing noise only. In the statistical filtering process, the variance value of each image segment will determine whether noise reduction is to be applied in

that area. Filters can thereby be designed that reduce noise without altering the object contours (Figs. 1A, B, E, and F).

Diffusion algorithms remove noise from an image by modifying the image via a partial differential equation. A simple implementation diffuses an image homogeneously by application of the diffusion equation (heat equation), which is equivalent to filtering the image with a Gaussian linear filter [17] with varying kernel size. In contrast, anisotropic diffusion filters control the diffusion process by an "edge stopping function" that depends on local image properties, i.e., the image gradient magnitude [12,18]. Thus, smoothing is applied at regions of homogeneous gray value intensities and prevented at edges. As a result of this process, noise is reduced and sharp edges are maintained in the image.

2.1.4. Segmentation by thresholding

Thresholding is a simple but often very efficient way to segment an image. This region-oriented segmentation technique classifies image points as object or background according to their intensity values. Thresholding is suitable to detect objects on a homogeneous background with a different gray value intensity than the objects. The threshold l assigns any image point (x, y) at $f(x, y) > l$ to objects, while points $f(x, y) < l$ are regarded as background (Figs. 1A–D). Unless the object in the image has extremely steep edges, the exact value of l can have considerable effect on the boundary position and thus the apparent size of the extracted object. Subsequent size measurements are thus rather sensitive to the threshold gray value. For this reason, one needs a consistent method to establish the threshold. An image containing an object on a contrasting background has a bimodal gray-level histogram. The two peaks correspond to the relatively large number of points inside and outside of the object. The dip between the peaks corresponds to the relatively few points around the edge of the object. This dip is commonly used to determine the threshold [17].

2.1.5. Confinement trees and connected operators

For a given level l, thresholding provides a binary image and assumes that each object is formed by a white region (with $f(x, y) > l$) entirely separated from other white pixels by black background pixels. Such regions are called confiners. Calculated for different levels l the confiners define a tree structure, called a confinement tree, as illustrated in Figs. 1K and L. The identification of each region (confiner) and the decision on whether it is related to a region in the previous or succeeding level in the confinement tree is computationally more demanding than the calculation of the binary image. However, an algorithm was proposed for calculating the confinement tree for all available gray levels of the image but requiring only a little more additional computation time than calculation of the binary image obtained by thresholding for one given threshold l [19]. Confiners can be deleted from the confinement tree according to certain filtering criteria [20,21].

Connected operators are closely related to confinement trees and can be obtained by filtering the tree according to some criteria (see above) followed by reconstruction of the image based on the filtered tree. In this reconstruction process the gray value of each pixel is defined depending on the levels for which the pixel is in a confiner that has been deleted or not. According to the filtering criteria, these operators can be "opening," "closing," "thinning," etc. [22]. They allow a more sophisticated and adapted choice for segmenting objects than simple thresholding with few additional computation costs.

2.1.6. Edge-based segmentation

Edge detection algorithms are used to establish the boundaries of objects within an image. First, each pixel and its immediate neighborhood are examined to determine whether the pixel is on the boundary of an object. Pixels that exhibit the required characteristics are labeled as edge points. An image with labeled edge points normally shows each object outlined in edge points, but generally not with closed connected boundaries. Thus, as a second step edge point linking is required to create closed connected boundaries.

If a pixel lies on the boundary of an object, its neighborhood is a zone of gray-level transition. The two characteristics of principal interest are the slope and the direction of that transition, that is, the magnitude and direction of the gradient vector. Edge detection operators are often convolution kernels (see Section 2.1.1) that implement directional derivatives, examining each pixel neighborhood, and quantify the slope and direction of the local gray-level transition. The nonmaximum suppression algorithm [12,23] determines candidate edge pixels if the gradient is maximal compared with the two neighbors in the direction of the gradient and if it has a potential predecessor and successor (Fig. 1G). Therefore, multiple responses to a single edge are suppressed and the formation of closed contours is assisted. To obtain closed borderlines, edges can be traced by taking into account local orientation (direction of the gradient) and equal probability of pixels belonging to adjacent regions (Fig. 1H). As a result of the tracing algorithm, closed borderlines enclosing homogeneous regions are obtained that can be used to build a region neighborhood graph [12]. Each node of the graph is associated with morphological parameters such as mean intensity, shape, and size according to the assigned region within the image. Objects can finally be detected by application of a selection criterion for these nodes, e.g., local maximal intensity (Figs. 1I and J).

2.1.7. Active contours

If the initial segmentation is suboptimal, it can often be improved by methods that refine the boundary initially assigned to an object by integrating global image features. The active contour or snake is a common tool for such refinement, which is particularly well suited when the object—moving globally and locally—has to be tracked through an image sequence because the contour (/surface) determined at the current time step can be used as initialization for the next step.

The active contour or snake has originally been introduced by Kass et al. [24]. An initial curve is put into the image according to the segmentation result of a previous time step or derived from an initial simpler segmentation method. Alternatively, it is placed interactively or automatically just roughly around the object's borders. An energy function is associated with the curve, which has an *internal* term penalizing stretching or bending and an *external* term assessing the fit into the image in each point of the curve. The external term can be calculated based on local properties of the image (e.g., gray level, gradient) or on previously extracted edges. During minimization of the energy function, the snake reacts to the image, and moves in a smooth, continuous manner toward the desired object boundary. Active contour techniques have also been used for the precise detection of object surfaces in 3D recordings, by minimizing an appropriate energy function expressing the quality of the fit of the deformable surface to the image [25–28]. Although snakes can accurately detect object boundaries in principle, a general limitation is the strong dependence on precise parameter settings that have to be determined for any specific application, especially when the 3D topology is complex and noise levels are high. Recent efforts have been made for automatic parameter adjustments and therefore snakes have become more generally applicable [27,28].

2.2. Motion estimation

In the center of a quantitative analysis of the motion in image sequences stands the estimation of the motion appearing in the sequence, if possible for each point and at each time step. In this section, we review three approaches to estimating the motion depending on the situation represented in the images. For tracking a large number of small particles that move individually and independently from each other, *single-particle tracking* approaches on previously segmented particles are most appropriate. Thus, often only the movement of the gravity center of a particle is considered but not the movement of the different points within the particle. For the determination of a more complex movement for each detected object or in the whole image scene two other approaches have been developed. On the one hand, *image registration* enables a computer to "register" (apprehend and allocate) certain objects in the real world as they appear in the computer's internal model. At first, only rigid mappings (rotation and translation) have been used to superimpose the images. Now, research is focused more on the integration of local deformations. On the other hand, techniques to estimate the local motion directly based on the pixel's gray values have been developed in image sequence analysis, a method referred to as *optical flow estimation*.

2.2.1. Single-particle tracking

For image sequences containing small objects with large displacements (with respect to the object's diameter) dynamic analysis can be achieved by single-particle tracking in time–space [29,30]. For each object in a given time frame corresponding objects in a previous time frame have to be found. Importantly, for an object in the current frame not necessarily the nearest object in the previous frame is the corresponding one ("correspondence problem"). Therefore, the correspondence is established based either on object features or on inter-object relationships of the objects inside a frame. In Hassan [31] the distances and the angles of the objects in the same frame have been taken into account. Object features can be dynamic criteria such as displacement and acceleration of an object as well as area/volume or mean gray value of the object. Assuming that optical flow is continuous (see Section 2.2.2), corresponding objects in subsequent images should be similar. Because noise sources during the imaging process distort this assumption, standard region-based matching techniques do not give satisfactory results [32]. A more reliable tracking approach involves fuzzy logic-based analysis of the tracking parameters [33].

Fuzzy theory assumes that all things are a matter of degree [34]. Fuzzy systems behave as associative memories mapping close inputs to close outputs without requiring a mathematical description of how the output functionally depends on the input. A fuzzy system relies on linguistic "rules" encoded in a numerical fuzzy associative memory mapping, the FAM rules. According to a dynamic particle model the velocity of an object is assumed to remain relatively constant. To compare two objects in consecutive images differences in velocity and deviation of expected extrapolated position from the potential new position are measured. In addition differences in total intensity and area are computed and translated into fuzzy rules.

2.2.2. Estimation of the local motion flow

The optical flow has been defined as the motion flow that can be derived from two consecutive images in a time series and is expressed by the motion vector field [17]. It is not equivalent to the real motion occurring in the image scene as, for instance, movements inside a

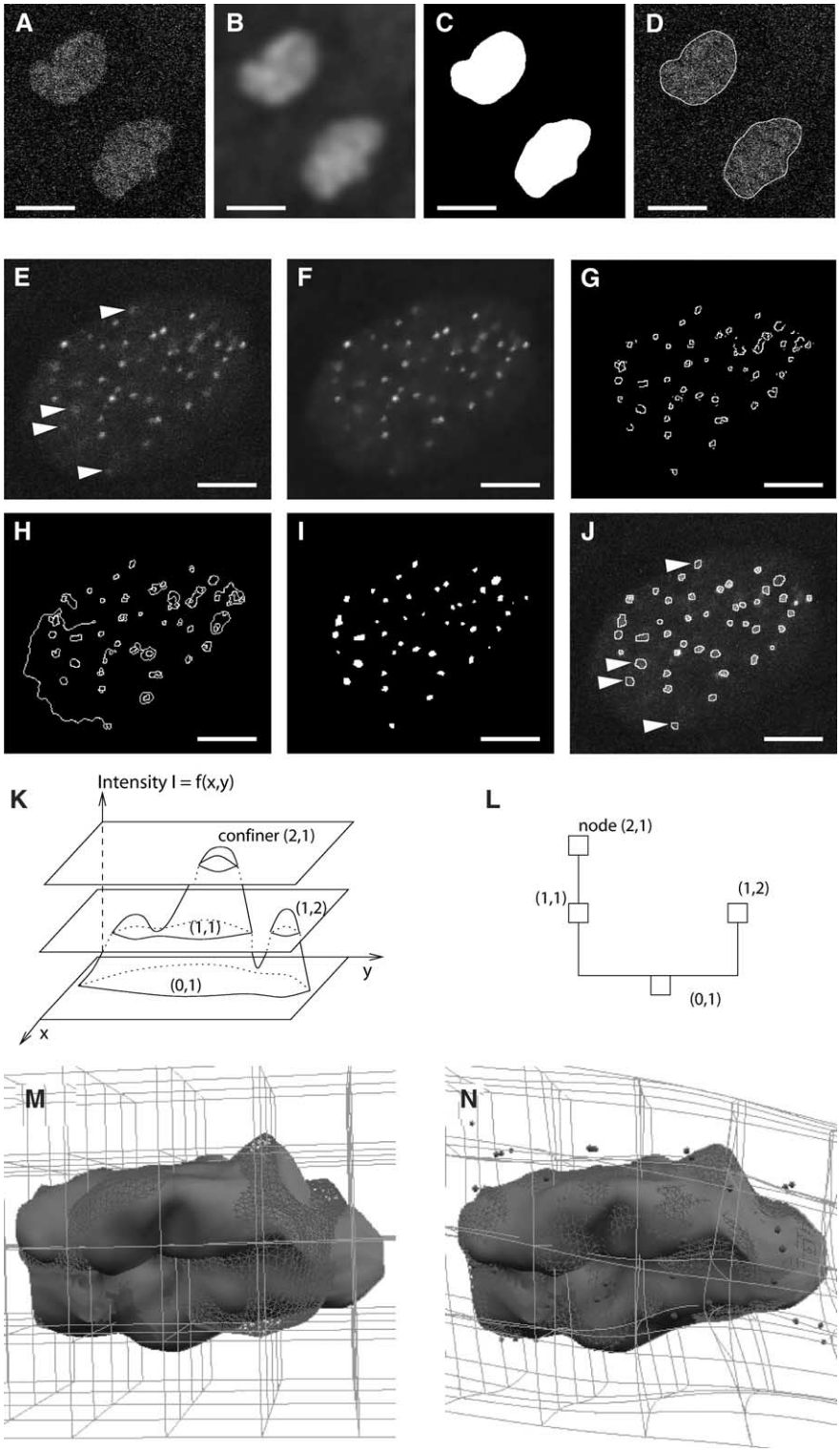


Fig. 1. Segmentation and motion estimation of fluorescently labeled structures in live cell recordings. (A–D) Segmentation of daughter nuclei in mitotic anaphase cells by thresholding. Normal rat kidney (NRK) cells expressing histone 2B fused to cyan fluorescent protein (H2B-CFP) were imaged on a confocal microscope setup as described in [13]. (A) Raw image. (B) Image after anisotropic diffusion filtering. (C) Binary image after thresholding of (B). (D) Contour of extracted objects overlaid on original image. (E–J) Edge-based segmentation of centromeres. HeLa cells expressing centromeric protein CENP-A fused to enhanced GFP (CENP–A–EGFP, [56]) were imaged on a confocal microscope. (E) Raw image: arrowheads mark centromeres with a low gray value. (F) Image after anisotropic diffusion filtering. (G) Candidate edge pixels detected by non-maximum suppression edge detection. (H) Closed contours after tracing of edges. (I) Regions extracted by selection of local maxima of mean gray value intensities. (J) Overlay of the object contours on the original image. Arrowheads mark same centromeres as in (E). (K, L) Definition of confiners (K) and the confinement tree (L). (M, N) Visualization of the extracted motion of an anaphase cell nucleus labeled with H2B-CFP by deformation grids (reproduced from [42]). Bars = 5 μm.

region of homogenous gray intensity cannot be detected. Classic approaches to estimate the optical flow [35] are based on the motion constraint equation (MCE), which is derived as a first-order Taylor development [36] from the equation expressing the conservation of the luminous intensity (called continuity of the optical flow): If $v(x,y,t)$ is the velocity of point (x,y) at time t (the displacement during 1 time unit) and f is the gray value function we have

$$\begin{aligned}\mathrm{DFD}(x,y,v,t) &= f((x,y)+v(x,y,t),t+1)-f((x,y),t)\\ &= 0.\end{aligned}$$

Here, $\mathrm{DFD}(x,y,v,t)$ is called the displaced frame difference (DFD). The MCE relates the spatial and the temporal derivatives of f linearly with v. However, it is valid only for small displacements. The latter is known under the notion "problem of large displacements." Moreover, MCE is an underdetermined equation (with respect to v) and either a so-called regularization has to be applied to relate the velocities in different points or a parametric approach as described in Section 2.2.3 has to be used. A synopsis of methods based on the hypothesis of intensity conservation is provided in [35].

2.2.3. Parametric image registration

A spatial transformation maps each point in the 2- or 3D space to another point in the same space. In the context discussed here, the considered transformations are given in a closed and parametric form; i.e., for a given parameter value a transformation is specified that allows the calculation, for each point in the space, of the transformed point by simple algebraic operations. An example is a rotation where, after specification of the rotation angles, the transformation of each point can be calculated by a matrix multiplication. A parametric image registration algorithm specifies the parameters of a given parameterized transformation in a way that physically corresponding points at two different time steps are brought as close as possible together. Such algorithms have been formulated in various ways: Some algorithms operate on previously extracted surface points [37,38] while others register the images directly based on the gray value differences and inbetween, features at a intermediary level, e.g., confiners or the confinement tree (Section 2.1.5, are extracted and indentified) [39]. Most commonly, a cost or error function is defined and an optimization method is chosen that iteratively adjusts the parameters until an optimum has been reached (minimum of the cost function). A classic similarity criterion based on extracted point features measures the Euclidean distance of nearest points in the two data sets. The squared sum of these distances can be minimized using the Levenberg–Marquardt optimization algorithm [37]. Furthermore, several intensity-based similarity measures, e.g., the sum of squared intensity differences, or the mutual information, are described in [40].

In cell biology, parametric registration has been used mainly for automated correction of rotational and translational movements in a time series. This allows enhancement of the visual interpretation of continuous time–space reconstructions by revealing only local dynamics, especially if the movement is a result of the superposition of two or more independent motions. It also makes estimation of the local dynamics more robust in this case. For example, when tracking particles inside the cell nucleus the global movements of the nucleus as a whole have to be compensated. The inverse case can be true as well: Only the global movements are of interest and the local movements are considered to be artifacts. In both cases *rigid* transformations need to be compensated for. When dealing with deformations, *nonrigid* registration algorithms have to be applied, which differ with respect to the so-called "motion model" [41,42] and the strategy to find the desired parameters varies.

2.3. Visualization

A detailed analysis of complex dynamic processes in cells would ideally be studied in three spatial dimensions over time. Such 4D imaging experiments can be performed e.g., on confocal or epifluorescence microscopes. Large and complex datasets, typically 5000–10,000 single images, are thereby generated, which cannot be analyzed appropriately without computational tools for the visual and quantitative inspection in space and time. Early studies have explored 4D datasets by simply browsing through an image gallery and by highlighting interactively selected structures [43]. To facilitate interpretation, many 4D experiments are preprocessed by a projection step reducing dimensionality (e.g., projection of image stacks to the x–y plane). A better interpretation of 4D imaging data can often be achieved with specialized computational image processing tools. Two commonly used rendering algorithms for displaying 3D structures are volume rendering and surface rendering by computer graphics [44].

2.3.1. Volume rendering

Volume rendering is a technique for visualizing complex 3D data sets without explicit definition of surface geometry. Volume visualization is achieved in three steps: classification, shading, and projection. The classification step assigns a level of opacity, contrast, and color to each voxel in the 3D volume [45,46]. Then, shading techniques are used to simulate both the object surface characteristics and the position and orientation of surfaces with respect to light sources and the observer. The colored, semitransparent volume is then projected onto a plane perpendicular to the observer's direction. Through each grid point on the projection

plane, a ray is cast into the volume. As the ray progresses through the volume, the color and opacity at evenly spaced sample locations are computed, finally yielding a single pixel color. While volume rendering techniques achieve satisfactory display of biological structures, this method is limited to pure visualization and does not deliver quantitative information. In addition, the high anisotropy typical of live cell imaging with low z-resolution limits the quality of this visualization technique (Fig. 2A).

2.3.2. Graphical surface rendering

Surface reconstructions approximate a selected structure by a list of polygons. The structure is displayed by projecting all the polygons onto a plane that is perpendicular to a selected viewing direction. The user can examine the displayed structure by changing the viewing direction interactively. Although the rendering algorithms are well developed, the generation of a polygon list that represents the surface appropriately can be difficult. The most commonly used method to triangulate the 3D surface is the Marching Cube algorithm [47]. The 3D structure is defined by a threshold value throughout the data set, constructing an isosurface. The drawback of this method is that the surface of many biological structures cannot be defined using a single intensity value, resulting in loss of relevant information. In addition, the anisotropic resolution characteristic of most optical microscopic data sets becomes very obvious at viewing angles that are perpendicular to the imaging axis. It is a major goal to enhance image quality by regaining spatial resolution in 4D live cell imaging experiments.

A recent computational approach was designed to deal particularly adequately with the high degree of anisotropy typical of 4D live cell recordings [13]. The visualization is based on a geometrical surface representation that is calculated from segmented image slices. Therefore, cellular structures have to be identified by segmentation first. Object identification is carried out in individual optical sections by edge-based segmentation techniques [12] or by thresholding. For the reconstruction of a continuous 3D surface from optical slices a parameterized contour running through all pixels of each segmented outline is required. Therefore, interpolation techniques are used that generate continuous curves from sampled contour points. Classic interpolation methods such as linear interpolation are not

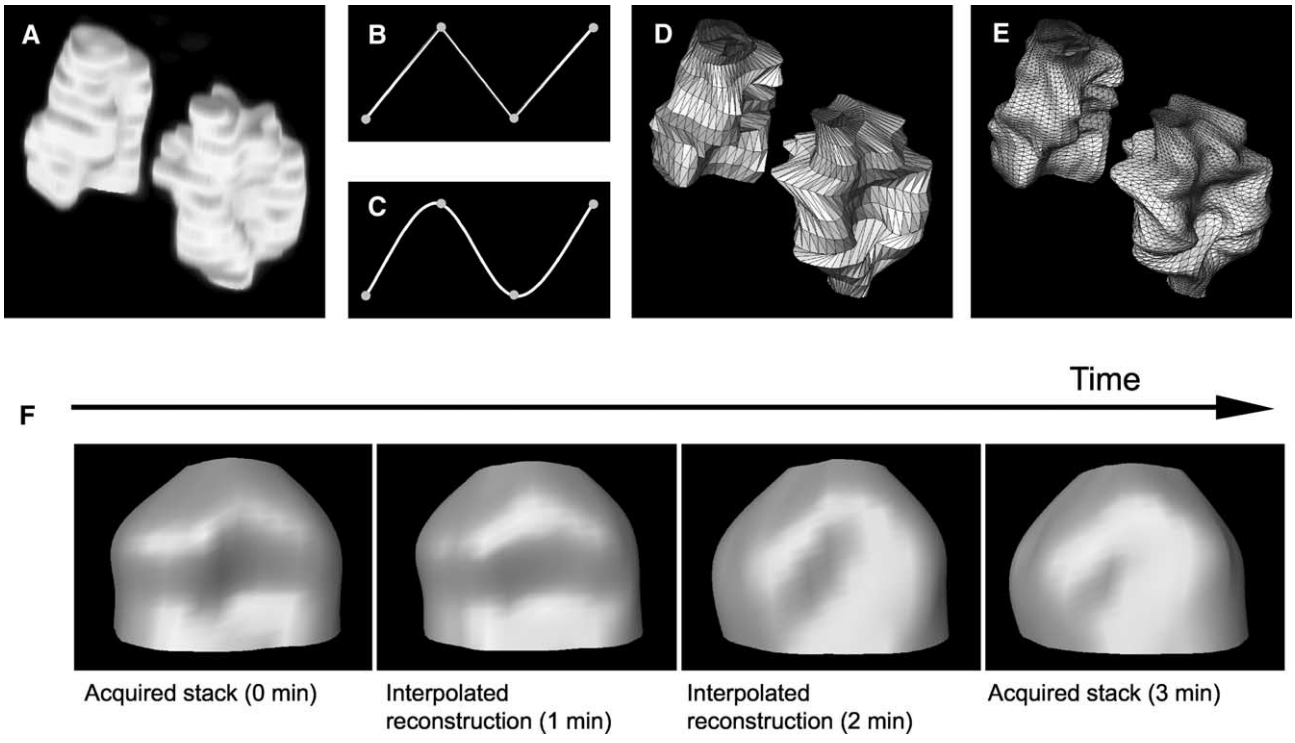


Fig. 2. Visualization of cellular structures. (A) Volume rendering of anaphase NRK cells expressing H2B-CFP. Eighteen z-slices were recorded on a confocal microscope with 1-μm steps. Volume rendering was carried out using the Amira 2.3 software (Template Graphics Software, San Diego, CA). (B) Linear interpolation between four outline points. (C) Interpolation between the same four outline points using cubic B-splines. (D) Visualization by surface rendering of the same data set as in (A). Outlines were detected in individual slices by anisotropic diffusion filtering and subsequent thresholding. Three-dimensional reconstructions were calculated by connecting corresponding outline points with linear interpolation, as described in [13]. (E) Similar to (D), but cubic B-splines were used to connect corresponding outline points. (F) Four-dimensional reconstruction of 3D stacks recorded with a time lapse of 3 min. Temporal intermediate reconstructions were obtained by cubic B-spline interpolation over time [13].

appropriate for the reconstruction of live cell imaging data, showing sharp edges at the sample points (Figs. 2B and D). In contrast, cubic splines allow generation of smooth curves or surfaces with second-order continuity (Figs. 2C and E). The idea of splines is to interpolate between points by a smooth curve with minimal bending, which is built up from segments connected in the given points. Splines are functions that are defined piecewise on intervals, in particular, as cubic polynomials, and fulfill smoothness criteria at the knots between individual segments. Cubic basic spline curves, often referred to as cubic B-splines, are a generalization of interpolating cubic polynomial splines and are more flexible with respect to changes of the curve. A detailed mathematical description of cubic B-splines can be found in [48].

Cubic B-splines were thus used to generate smooth curves approximating the contour pixels in individual optical slices. In the next step, equally parameterized contour points from adjacent z slices were connected, again by using cubic B-splines. From these curves, intermediate contours between optical slices were obtained to increase spatial resolution of the surface model and avoid sharp edges (Figs. 2D and E). The specific advantage of an interpolated surface rendering is a very distinct display of small-scale features, as compared with volume rendering (Fig. 2A). Moreover, surface reconstruction allows direct access to quantitative data (see Section 2.4).

2.3.3. Visualization of 4D data

A sequence of 3D reconstructions can be used to visualize a 4D dataset. However, besides the number of z sections per stack, the temporal resolution is limited in live cell experiments. Thus, with the surface reconstructions of cellular structures at distinct time points at hand, it was of great interest to infer continuous motion from these data. Interpolation over time is especially useful when the experiment is carried out over longer time courses, thus enlarging the time-lapse interval between consecutive image stacks.

To achieve temporal interpolation, an algorithm has to be developed for smooth transition from one surface reconstruction to the next. Methods for this task are referred to as morphing. The morphing problem consists of two major tasks. For objects that are represented as point meshes, such as generated by the 3D surface reconstruction, corresponding points have to be identified in subsequent time frames. Such correspondence can be obtained as a result of motion estimation (Section 2.2). In the absence of a motion estimation step correspondence has been established using identical parameterization in subsequent frames [25]. In the next step, interpolation over time between these corresponding surface points is carried out. B-Splines were used to enhance temporal resolution by interpolation. A continuous reconstruction of entire 4D data sets was achieved (Fig. 2F and see supplementary video material in [13]). The animated surface reconstruction was visualized in a multidimensional virtual reality viewer that allows real-time user interaction (OpenInventor SceneViewer, Template Graphics Software, San Diego, CA).

2.3.4. Visualization of the quantified motion

Several techniques have been used to depict the motion occurring in the image sequence after quantitative evaluation. Classically, motion vectors regularly placed on the image, trajectories, or deformed grids give the user an impression of the occurring motion. Whereas motion vectors provide more details, deformed grids allow one to apprehend the global movement and the bending of the space. An illustration of a deformed volume grid rendered in three dimensions as proposed in [44] is shown in Fig. 1N. To have a continuous visualization of the motion in specific points, trajectories are well suited (Fig. 3A and B; [6,7,12,49]. Techniques to visualize scalar quantitative values associated with each point on a surface or in space use color, gray intensity, or patterns [50]. They can be combined with the techniques mentioned before. The statistical analysis of the extracted quantitative values comes with further possibilities of presenting these values.

2.4. Quantitative analysis

A great advantage of combined segmentation and surface reconstruction is the immediate access to quantitative information that corresponds to visual data [13]. The binarized object representation can be used to directly measure volume over time. Moreover, the gray values in the segmented area of corresponding original images can be measured to determine the amount and concentration of fluorescently labeled protein in the segmented cellular compartments. After motion estimation, the velocity of the detected object mass center or, alternatively, of each point on the object surface or even of each point in space is available. With this, acceleration [6], tension or bending [51], or diffusion coefficients [4,8,9,12,52] can be determined. Statistical analysis provides further possibilities of motion characterization. For instance, the peaks of velocity histograms report which velocity values occur most frequently [53]. Also, the movements of different objects (or of the same object at different time steps) can be compared by their velocity histograms. A challenge for future work is to extract more specific parameters by fitting a biophysical model to the data. This has been achieved in medical image analysis [50], where the human brain was modeled using finite element methods, giving insight into forces occurring during brain deformations.

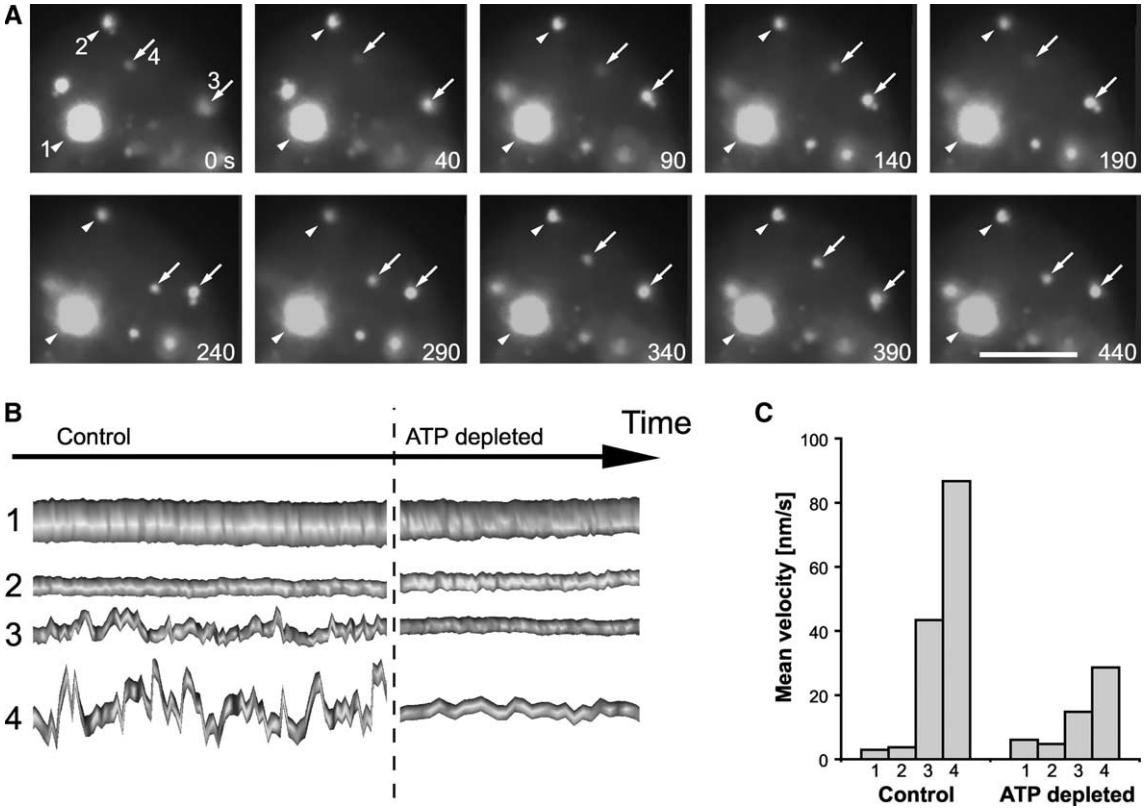


Fig. 3. Quantitative analysis of metabolic energy-dependent dynamics of PML bodies. Baby hamster kidney cells expressing Sp100 protein fused to enhanced yellow fluorescent protein (EYFP-Sp100) were imaged on a wide-field fluorescence microscope with a time lapse of 10 s [7]. (A) Selected frames from the live cell recording. Arrows mark PML bodies that move with high velocities; arrowheads depict more static PML bodies. (B) Graphical display of the trajectories from the PML bodies as numbered in (A). PML bodies were detected by anisotropic diffusion filtering and thresholding, and tracked over time using fuzzy logic-based algorithms [12]. Trajectories were visualized as a graphical time–space reconstruction [12]. Control cells were imaged, then perfused with 6 mM sodium azide to deplete ATP. (C) Mean velocities of the trajectories as numbered in (A) before (control) and after ATP depletion. Bar = 5 μm.

3. Applications

In vivo imaging with GFP-tagged constituents of various nuclear subcompartments has revealed their dynamic organization in the interphase nucleus. A particularly striking example is the dynamics of nuclear speckles that consist mainly of nuclear RNA splicing factors. Nuclear speckles can undergo regulated reorganization of their morphology and dynamics [3,6]. Under normal conditions these nuclear speckles grossly alter their surface morphology, and can show budding of fragments and fusion to other nuclear speckles. On treatment with the transcription inhibitor α-amanitin, speckles round up and no budding or fusion occurs. The computer methods for image analysis described in this review have been used to quantitatively investigate regulated dynamics of nuclear speckles [6]. This confirmed the results obtained from visual inspection, and allows quantitative comparison with imaging data generated in other laboratories.

Many studies on nuclear architecture have now shown that other nuclear subcompartments show considerable dynamics in the interphase nucleus, including Cajal bodies [5,10], PML bodies [7], and chromatin [4,8,9]. Quantitative image analysis showed that chromatin underlies slow diffusional motion [4,8,9,54], and this movement is confined to relatively small regions in the nucleus. Importantly, the constraint on diffusional motion is regulated throughout the cell cycle [8,9]. A long-standing question has been whether nuclear compartments can also undergo directed, energy-dependent movements, thereby providing a potential mechanism of regulated gene expression. Quantitative live cell imaging could now establish for PML bodies, Cajal bodies, and chromatin that such transport dependent on metabolic energy can occur in the nucleus (Fig. 3) [7,9,10].

4. Conclusions

In this review, we have summarized concepts of quantitative image analysis of multidimensional microscopy data. In particular, we focus on recent

approaches for image segmentation, motion estimation, and visualization that were designed to suit the specific demands of live cell microscopy (e.g., low signal-to-noise, high anisotropy). The applicability of the described methods is demonstrated by a number of applications on experimental data. Most importantly, we point out that objectivity in the interpretation of dynamic imaging data can be achieved only through automated computational tools, such as described in this review. Quantitative tools for measurements of velocity, acceleration, diffusion coefficients, volume changes, concentrations, and distances in 4D datasets are now available.

Moreover, quantitative data from live cell microscopy can be exploited to generate mathematical models that describe biological processes. Mathematical models are important tools for quantitative hypothesis testing and can support the design of further experiments [55]. In the future, modeling of cell biological processes will certainly gain importance, since it provides a means to integrate observations from many individual experiments to a complex system. The quantitative imaging methods presented here provide an important building block for the description of biological phenomena on a systems level, allowing the modeling of structural changes in cells. Furthermore, a mathematical description of whole biological systems also includes models of genetic networks, interaction networks of proteins, metabolic processes, and signal transduction networks. It is hoped that this general strategy, termed the systems biology approach, will allow identification of novel principles of cellular regulation in the huge amount of experimental data that are currently generated.

5. Software packages

5.1. TILL photonics: visTRAC

Software package for recording and quantitative analysis of time-resolved cellular processes on fluorescence microscopes. Features include automated object identification with anisotropic diffusion filtering and edge-based segmentation, single-particle tracking based on fuzzy-logic algorithms, graphical visualization in an interactive virtual reality viewer, and quantitative analysis of dynamic parameters. For more information see: http://www.till-photonics.de.

5.2. TGS, indeed visual concepts: Amira

Powerful 3D visualization toolbox including volume rendering and graphical surface rendering techniques. Basic tools for quantitative analysis available. For more information see: http://www.tgs.com.

5.3. NIH: ImageJ

Free software for quantitative image analysis. Many specialized plug-ins available. Additional modules can be programmed by using the Java interface. More information and software download at: http://rsb.info.nih.gov/ij/.

5.4. Zeiss: LSM 5

Software for control of confocal microscope and quantitative image analysis. Three-dimensional module available for graphical display of 4D imaging data. More information at: http://www.zeiss.com.

5.5. Bitplane AG: Voxelshop pro, Imaris, surface

Voxelshop pro is a module to separate and quantify automatically objects characterized by gray levels or texture. *Imaris* is a high-quality software package used to process and visualize 3D images. It has been designed to accept most microscopic image formats and offers a range of functions. *Surface* automatically converts a volume image into a geometric object made of triangles. See: http://www.bitplane.com.

5.6. SIS: AnalySIS

Compatible with most cameras and microscopy for image acquisition and import, image display and editing, archiving, documentation, image processing, measuring, analysis, including a macro scripting language. See: http://www.soft-imaging.net.

5.7. AnalyzeDirect: Analyze

Analyze includes volume rendering, virtual endoscopy, segmentation, image registration, surface rendering, measurements, and many other imaging functions. See: http://www.analyzedirect.com.

Acknowledgments

The authors thank D.L. Spector and M. Muratani for kindly providing imaging data on PML body dynamics, J. Ellenberg and J. Beaudouin for providing imaging data of mitotic nuclei, K.F. Sullivan for the generous gift of CENP-A-EGFP, and N. Daigle for helpful comments on the manuscript.

References

[1] M. Chalfie, Y. Tu, G. Euskirchen, W.W. Ward, D.C. Prasher, Science 263 (1994) 802–805.

[2] P. Heun, A. Taddei, S.M. Gasser, Trends Cell Biol. 11 (2001) 519–525.
[3] T. Misteli, J.F. Caceres, D.L. Spector, Nature 387 (1997) 523–527.
[4] W.F. Marshall, A. Straight, J.F. Marko, J. Swedlow, A. Dernburg, A.S. Belmont, A.W. Murray, D.A. Agard, J.W. Sedat, Curr. Biol. 7 (1997) 930–939.
[5] M. Platani, I. Goldberg, J.R. Swedlow, A.I. Lamond, J. Cell. Biol. 151 (2000) 1561–1574.
[6] R. Eils, D. Gerlich, W. Tvarusko, D.L. Spector, T. Misteli, Mol. Biol. Cell. 11 (2000) 413–418.
[7] M. Muratani, D. Gerlich, S.M. Janicki, M. Gebhard, R. Eils, D.L. Spector, Nat. Cell. Biol. 4 (2002) 106–110.
[8] J. Vazquez, A.S. Belmont, J.W. Sedat, Curr. Biol. 11 (2001) 1227–1239.
[9] P. Heun, T. Laroche, K. Shimada, P. Furrer, S. Gasser, Science 7 (2001) 2181–2186.
[10] M. Platani, I. Goldberg, A.I. Lamond, J.R. Swedlow, Nat. Cell. Biol. 4 (2002) 502–508.
[11] F. Germain, A. Doisy, X. Ronot, P. Tracqui, in: IEEE Trans. Biomed. Eng., 46, 1999, pp. 584–600.
[12] W. Tvaruskó, M. Bentele, T. Misteli, R. Rudolf, C. Kaether, D.L. Spector, H.H. Gerdes, R. Eils, Proc. Natl. Acad. Sci. USA 96 (1999) 7950–7955.
[13] D. Gerlich, J. Beaudouin, M. Gebhard, J. Ellenberg, R. Eils, Nat. Cell. Biol. 3 (2001) 852–855.
[14] L.G. Brown, ACM Comput. Surveys 24 (1992) 325–376.
[15] S. Lavallée, in: R. Taylor, S. Lavallée, G. Burdea, R. Mösges (Eds.), Computer Integrated Surgery, MIT Press, Cambridge, MA, 1996, pp. 77–97.
[16] S.C. Zhu, A.L. Yuille, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1996) 884–900.
[17] B. Jahne, Digital Image Processing—Concepts, Algorithms, and Scientific Applications, Springer-Verlag, Berlin, 1997.
[18] M.J. Black, G. Sapiro, D. Marimont, D. Heeger, IEEE Trans. Image Process. 7 (1998) 421.
[19] J. Mattes, J. Demongeot, Lect. Notes Comput. Sci. 1953 (2000) 392–405.
[20] P. Salembier, A. Oliveras, L. Garrido, IEEE Trans. Image Process. 7 (1998) 555–570.
[21] R. Jones, Comput. Vision Image Understanding 75 (1999) 215–228.
[22] P. Soille, Morphological Image Analysis—Principles and Applications, Spring-Verlag, Berlin, 1999.
[23] R. Nevatia, K.R. Babu, Comput. Graphics Image Process. 13 (1980) 257–269.
[24] M. Kass, A. Witkin, D. Terzopoulos, J. Comput. Vision 1 (1988) 321–331.
[25] F. Leitner, I. Marque, S. Lavallée, P. Cinquin, in: Curves and Surfaces, Chamomix, France, 1990, pp. 279–284.
[26] C. Xu, D.L. Pham, J.L. Prince, in: J.M. Fitzpatrick, M. Sonka (Eds.), Medical Image Processing and Analysis, vol. 2, 2000, pp. 129–174.
[27] M. Gebhard, J. Mattes, R. Eils, in: MICCAI, vol. 2208, 2001, pp. 1373–1375.
[28] M. Gebhard, R. Eils, J. Mattes, in: International Conference on Diagnostic Imaging and Analysis, Shanghai, China, 2002, pp. 125–130.
[29] I. Grant, Selected Papers on Particle Image Velocimetry, SPIE Press, 1994.
[30] L. Adamczyk, A.A. Rimai, Exp. Fluids 6 (1988) 373–380.
[31] Y. Hassan, Exp. Fluids 12 (1991) 49–60.
[32] P. Anandan, Int. J. Comput. Vision 2 (1989) 283–310.
[33] F. Hering, C. Leue, D. Wierzimok, B. Jähne, Exp. Fluids 23 (1997) 472–482.
[34] B. Kosko, Neural Networks and Fuzzy Systems, Prentice Hall, Engelwood Cliffs, NJ, 1992.
[35] A. Mitiche, P. Bouthemy, Int. J. Comput. Vision 19 (1996) 29–55.
[36] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C—The Art of Scientific Computing, Cambridge University Press, Cambridge, 1992.
[37] S. Lavallée, R. Szeliski, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 378–390.
[38] P.J. Besl, N.D. McKay, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 239–256.
[39] J. Mattes, J. Demongeot, in: SPIE Medical Imaging 2001: Image Processing, San Diego, vol. 4322, 2001, pp. 602–610.
[40] A. Roche, G. Malandain, N. Ayache, S. Prima, in: C. Taylor, A. Colchester (Eds.), MICCAI'99, vol. 1679, Springer-Verlag, Cambridge, UK, 1999, pp. 555–566.
[41] J. Mattes, J. Fieres, R. Eils, in: SPIE Medical Imaging 2002: Image Processing, San Diego, vol. 4684, 2002, pp. 518–527.
[42] J. Fieres, J. Mattes, R. Eils, in: DAGM, Springer-Verlag, Berlin, vol. 2191, 2001, pp. 76–83.
[43] C. Thomas, P. DeVries, J. Hardin, J. White, Science 273 (1996) 603–607.
[44] H. Chen, J.R. Swedlow, M. Grote, J.W. Sedat, D.A. Agard, in: J.B. Pawley (Ed.), Handbook of Biological Confocal Microscopy, Plenum Press, New York, 1995, pp. 197–210.
[45] S.J. Wright, V.E. Centonze, S.A. Stricker, P.J. DeVries, S.W. Paddock, G. Schatten, Methods Cell. Biol. 38 (1993) 1–45.
[46] R.F. Gasser, S. Shigihara, L.F. Camero, Comput. Med. Imaging Graphics 20 (1996) 449–457.
[47] H.E. Cline, W.E. Lorensen, S. Ludke, C.r. Crawford, B.C. Teeter, Med. Phys. 15 (1988) 320–327.
[48] A. Watt, W.M. Watt, Advanced Animation and Rendering Techniques, Addison-Wesley, New York, 1992.
[49] A. Rustom, D. Gerlich, R. Rudolf, C. Heinemann, R. Eils, H.H. Gerdes, Biotechniques 28 (2000) 722–728, See also p. 730.
[50] M. Ferrant, A. Nabavi, R. Kikinis, S.K. Warfield, in: SPIE Medical Imaging, San Diego, vol. 4319, 2001, pp. 366–373.
[51] F. Bookstein, IEEE Trans. Pattern Anal. Mach. Intel. 11 (1989) 567–585.
[52] J.R. Chubb, S. Boyle, P. Perry, W.A. Bickmore, Curr. Biol. 12 (2002) 439–445.
[53] D. Uttenweiler, C. Veigel, R. Steubing, C. Götz, S. Mann, H. Haussecker, B. Jähne, R. Fink, Biophys. J. 78 (2000) 2709–2715.
[54] H. Bornfleth, P. Edelmann, D. Zink, T. Cremer, C. Cremer, Biophys. J. 77 (1999) 2871–2886.
[55] R.D. Phair, T. Misteli, Nat. Rev. Mol. Cell. Biol. 2 (2001) 898–907.
[56] K. Sugimoto, R. Fukuda, M. Himeno, Cell. Struct. Funct. 25 (2000) 253–261.

Motion Analysis of Biomedical Objects

Alfredo López, Julian Mattes

Institute of Biomedical Image Analysis, UMIT Hall in Tirol, Autria

Biomedical Data Analysis Group, SCCH, Austria

The Motion of the Heart during the cardiac cycle is complex to describe and involves the interaction of several different structures. The analysis of the cardiac movement has various clinical applications and additionally, most of the techniques developed for the heart can be extended to the study of other biomedical objects.

Statistical motion models involve knowledge about the statistical distribution of the motion in already investigated processes related to object within the same population. Such models allow, for example, determining if specific motion patterns are associated with certain heart diseases (cluster analysis) or the extraction of the more significative motion-types that describe the movement of a set of subjects (principal component analysis).

The aim of our research in the frame of the EU-alfa project IPECA is to develop and evaluate new strategies in this research field. Recently, some few attempts have been made to build 4D probabilistic atlases which capture both the anatomical and functional variability of biomedical objects across a group of subjects [PLC04]. This issue is achieved by spatio-temporal registration methods which correct spatial and temporal misalignment of image sequences [PMR05]. Due to the restriction of the motion by the model usually the real motion in two consecutives images is only detected up to a residual error which accumulates when following the motion during the sequence [MEF01]. A further problem which has to be addressed is how far the spatio-temporal registration is able to resolve spatio-temporal ambiguities. These occur when it is not clear whether the difference between image sequences is due to spatial or/and temporal domain misalignment. Also, topological changes occurring in the image sequence such as fusion or separation of two objects are difficult to deal with. Whereas for the statistical representation of static shape [CTC95, DTC02, PFS04] more and more experience is available now, for the representation of motion questions as “which parameters shall be represented by the basis of the field space?” or “is it reasonable to decompose the spatial and temporal domain?” are still to clarify.

The aim our research is to develop and evaluate strategies to tackle some of the problems mentioned above. To validate the investigated model based approach a comparison between parametric and non-parametric approaches [RPP04] (see also classical pixel-wise optical flow approaches [HoS81, GuP95]) would be of interest. Emphasis should be placed on the value for the investigated application. For instance, using the statistical information it should be analyzed which component of the heart’s motion is due to breathing. Data for a number of hearth cycles as well as for moving cell nuclei are available already.

[MHA02] McLesh K., Hill D.L.G., Atkinson D., Blackball J.M., Razavi R. A study of the motion of and deformation of the heart due to respiration. IEEE Trans. Med Imag. 21 (2002) 1142-1150

[MFE02] J. Mattes, J. Fieres and R. Eils: A shape adapted motion model based on thin plate splines and pont clustering for point set registration. In: *SPIE Medical Imaging 2002: Image Processing,* San Diego, 23.-28. Feb. 2002, Proceedings of SPIE 4684 (518-527)

[CMR04] R. Chandrashekara, R.H. Mohiaddin, and D. Ruekert. Analysis of 3-D myocardial motion in tagged MR images using nonrigid image registration. IEEE Trans. Med Imag. (2004) 1245-1250

[BKP98] Blackball J.M., King A.P., Penney G.P., Addam A., Hawkes D.J.: A Statistical Model of Respiratory Motion and Deformation of the Liver. In: MICCAI'01. *LNCS* 2208 (2001) 1338-1340

[MEF01] J. Mattes et al.. Quantitative Analyse, Visualisierung und Bewegungskorrektur in dynamischen Prozessen, 2001, (German Patent Pending 101 11 226.2, 101 44 629.2, PCT Patent Application)

[CTC95] Cootes T.F., Taylor C.J., Cooper D.H., Graham J.: Active shape models – their training and their applications. *Computer Vision and Image Understanding* 61 (1995) 38-59

[DTC02] Davies RH, Twining CJ, Cootes TF, Waterton JC, Taylor CJ, "A Minimum Description Length Approach to Statistical Shape Modelling, IEEE Trans Med. Imaging, Vol 21 (2002), 525-537, 2002.

[PFS04] PilgramR , Fritscher KD, Schubert R, "Modeling of the geometric variation and analysis of the right atrium and right ventricle motion of the human heart using PCA", CARS 2004 - Computer Assisted Radiology and Surgery. Proceedings of the 18th International Congress and Exhibition, Vol. 1268C, 1108-1113, 2004.

[CRS03] R. Chandrashekara, A. Rao, G. I. Sanchez-Ortiz, R. H. Mohiaddin, and D. Rueckert. Construction of a statistical model for cardiac motion analysis using non-rigid image registration. In Information Processing in Medical Imaging: Proc. 18th International Conference (IPMI '03), Lecture Notes in Computer Science, pages 599-610, 2003.

[RPP04] N. Rougon, C. Petijean, F.Preteux
Building and using a statistical 3D motion atlas for analyzing myocardial contraction in MRI Proceedings SPIE Conference on Image Processing - SPIE International Symposium Medical Imaging'04, San Diego, CA, Vol. 5370, 14-19 February 2004, p. 253-264.

[PLC04] D. Perperidis, M. Lorenzo-Valdes, R. Chandrashekara, A. Rao, R. Mohiaddin, G. I. Sanchez- Ortiz, and D. Rueckert. Building a 4D atlas of the cardiac anatomy and motion using MR imaging. In IEEE International Symposium on Biomedical Imaging, pages 412-415, 2004.

[PMR05] Dimitrios Perperidis, Raad H. Mohiaddin and Daniel Rueckert Spatio-temporal free-form registration of cardiac MR image sequences. Medical Image Analysis, Vol. 9(5), October 2005, p. 441-456

[HoS81] Horn B., Schunck B.G.: Determining optical flow. Art. Intel. 17 (1981) 185-203

[GuP95] Gupta S.N. and Prince J.L.: On variable brighteness optical flow for tagged MRI. In: Inform. Proc. Med. Imag. (1995) 323-334

Towards Robust Rank Correlation Measures for Numerical Observations on the Basis of Fuzzy Orderings

Ulrich Bodenhofer
Institute of Bioinformatics
Johannes Kepler University Linz
4040 Linz, Austria
bodenhofer@bioinf.jku.at

Frank Klawonn
University of Applied Sciences
Braunschweig/Wolfenbüttel
38302 Wolfenbüttel, Germany
f.klawonn@fh-wolfenbuettel.de

Abstract

This paper aims to demonstrate that established rank correlation measures are not ideally suited for measuring rank correlation for numerical data that are perturbed by noise. We propose a robust rank correlation measure on the basis of fuzzy orderings. The superiority of the new measure is demonstrated by means of illustrative examples.

Keywords: Fuzzy Orderings, Rank Correlation, Robust Statistics.

1 Introduction

Correlation measures are among the most basic tools in statistical data analysis and machine learning. They are applied to pairs of observations ($n \geq 2$)

$$(x_i, y_i)_{i=1}^n \tag{1}$$

to measure to which extent the two observations comply with a certain model. The most prominent representative is surely *Pearson's product moment coefficient* [1, 14], often nonchalantly called *correlation coefficient* for short. Pearson's product moment coefficient is applicable to numerical data and assumes a linear relationship as the underlying model; therefore, it can be used to detect linear relationships, but no non-linear ones.

Rank correlation measures [9, 11, 13] are intended to measure to which extent a monotonic function is able to model the inherent relationship between the two observables. They neither assume a specific parametric model nor specific distributions of the observables. They can be applied to ordinal data and, if some ordering relation is given, to numerical data too. Therefore, rank correlation measures are ideally suited for detecting monotonic relationships, in particular, if more specific information about the data is not available. The two most common approaches are *Spearman's rank correlation coefficient* (short *Spearman's rho*) [16, 17] and *Kendall's tau (rank correlation coefficient)* [2, 10, 11].

This paper argues why the well-known rank correlation measures are not ideally suited for measuring rank correlation for numerical data that are perturbed by noise. Consequently, we propose a robust rank correlation measure on the basis of fuzzy orderings. The superiority of the new measure is demonstrated by means of illustrative examples.

2 An Overview of Rank Correlation Measures

Assume that we are given a family of pairs as in (1), where all x_i and y_i are from linearly ordered domains X and Y, respectively. *Spearman's rho* is computed as

$$\rho = 1 - 6\frac{\sum_{i=1}^n (r(x_i) - r(y_i))^2}{n(n^2-1)},$$

where $r(x_i)$ is the rank of value x_i if we sort the list $(x_1, \ldots, x_n)$; $r(y_i)$ is defined analogously. So, Spearman's rho measures the sum of quadratic distances of ranks and scales this measure to the interval $[-1, 1]$. It can be checked easily that a value of 1 is obtained if the two rankings coincide and that a value of -1 is obtained if one ranking is the reverse of the respective other. Note that the above definition of $r(x_i)$ and $r(y_i)$ was simplified, because it did not take coinciding values, so-called *ties*, into account. In such a case, the values $r(x_i)$ are usually defined as the mean value of all ranks of consecutive coinciding values in the sorted list.

With the same assumptions as above, *Kendall's tau* is computed as the quotient

$$\tau_a = \frac{C - D}{\frac{1}{2}n(n-1)},$$

where C and D denote the numbers of *concordant* and *discordant pairs*, respectively:

$$C = |\{(i,j) \mid x_i < x_j \text{ and } y_i < y_j\}|$$
$$D = |\{(i,j) \mid x_i < x_j \text{ and } y_i > y_j\}|$$

As above, if we have no ties and the two rankings coincide, we have $\frac{1}{2}n(n-1)$ concordant and no discordant pairs, so $\tau_a = 1$; if we have no ties and one ranking is the reverse of the respective other, we have no concordant and $\frac{1}{2}n(n-1)$ discordant pairs, so a value of $\tau_a = -1$ is obtained.

In the above definition of τ_a, ties, no matter whether in the first or in the second list, are not counted. So ties lower the absolute value of τ_a. Therefore, τ_a is best suited for detecting strictly monotonic relationships, but not ideally suited in the presence of ties. A well-established second variant [11],

$$\tau_b = \frac{C - D}{\sqrt{\frac{1}{2}n(n-1) - T}\sqrt{\frac{1}{2}n(n-1) - U}},$$

where

$$T = |\{(i,j) \mid x_i = x_j\}|, \quad U = |\{(i,j) \mid y_i = y_j\}|,$$

takes ties into account, but is still not fully robust to ties. A simple and tie-robust rank correlation measure is the *gamma rank correlation measure* according to Goodman and Kruskal [9] that is defined as

$$\gamma = \frac{C - D}{C + D}.$$

3 Motivation

All rank correlation measures highlighted above have been introduced with the aim to measure rank correlation of ordinal data (e.g. natural numbers, marks, quality classes, ranks). The measurement of rank correlation for *real-valued data*, however, is equally important in statistics and machine learning, but raises completely new issues. Depending on the source, numerical data are almost always subject to random perturbations—noise. The concepts introduced above do not take this into account. Pairs are counted as concordant or discordant only on the basis of ordering relations, but without taking into account that only minimal differences may decide whether a pair is concordant or discordant. If one observable depends on the other in a clearly monotonic way and if the level of noise is low, then the rank correlation measures introduced above will still reveal this strictly monotonic relationship and will not be compromised by minor local effects of noise. In the presence of a larger percentage of ties, however, already the slightest perturbations may lead to situations in which the above rank

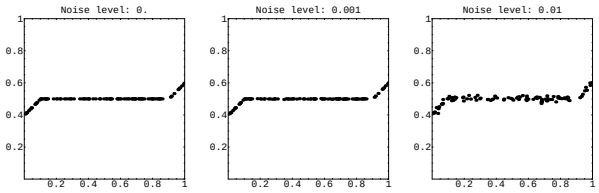


Figure 1: Scatter plots of a simple monotonic relationship with different noise levels.

correlation coefficients cannot yield meaningful results anymore. Consider the data sets in Figure 1. We see a monotonic, yet not strictly monotonic, relationship. The left plot shows data without noise, i.e. $y_i = f(x_i)$ for a non-decreasing function f. For these data, we obtain $\rho = 0.737$, $\tau_b = 0.639$ and $\gamma = 1$ (which confirms that γ is most robust to ties). The middle plot shows the same data, but with additive normally distributed noise with zero mean and $\sigma = 0.001$. Although it is hard to see the noise at all, we obtain $\rho = 0.519$ and $\tau_b = \gamma = 0.387$. These results indicate that none of the three measures can adequately handle a large proportion of ties in the presence of noise. For $\sigma = 0.01$ (right plot), the values are slightly lower, but not significantly: $\rho = 0.456$ and $\tau_b = \gamma = 0.331$. So we can conclude that it is rather the presence of noise in general than the magnitude of noise that distracts the three rank correlation measures.

The obvious reason for the weakness described above is the fact that all measures only take ordering relationships into account, but neglect similarities of data points. To illustrate that, consider the two pairs (a, c) and (b, c), where $b > a$. Obviously, this is a tie in the second component. If we add some noise to the second component of the second pair, i.e., if we replace (b, c) by $(b, c + \varepsilon)$, then ε decides whether $((a, c), (b, c + \varepsilon))$ is a tie (for $\varepsilon = 0$), concordant ($\varepsilon > 0$), or discordant $\varepsilon < 0$), where the magnitude of ε plays no role at all. So we observe a discontinuous behavior. This toy example thereby serves as a proof that all measures introduced above depend on the data in a discontinuous way.

The question arises how we can define a robust rank correlation measure that depends continuously on the data by taking similarities into account, but still serves as a meaningful measure of rank correlation. Obviously, the measure should be designed such that close-to-tie pairs receive less attention than pairs that are clearly concordant or discordant. A reasonable idea would be to base such a concept on the probabilities to which concordant/discordant pairs are observed as such compared to the probabilities that they are falsely observed as something else. That may be a reasonable approach. Note, however, that such probabilities can

only be computed if we know the joint distribution of x and y values or at least if we make distribution assumptions. In practice, such information is most often unavailable and, surely, we do not want to sacrifice the unique feature of rank correlation measures that they are *distribution-free*.

In our opinion, *fuzzy orderings* provide a meaningful way to overcome the difficulties explained above.

4 Fuzzy Orderings

Before we can introduce a fuzzy ordering-based rank correlation coefficient, we need to provide some basics of fuzzy orderings. We restrict to an absolutely necessary minimum and refer to literature for details. We assume that the reader is aware of the most basic concepts of triangular norms and fuzzy relations.

A fuzzy relation $L : X^2 \to [0,1]$ is called *fuzzy ordering* with respect to a t-norm T and a T-equivalence E, for brevity *T-E-ordering*, if and only if it is T-transitive and fulfills the following two axioms for all $x, y \in X$:

(i) E-Reflexivity: $E(x,y) \leq L(x,y)$

(ii) T-E-antisymmetry: $T\big(L(x,y), L(y,x)\big) \leq E(x,y)$

Moreover, we call a T-E-ordering L *strongly complete* if $\max\big(L(x,y), L(y,x)\big) = 1$ for all $x, y \in X$ [4].

Several correspondences between distances and fuzzy equivalence relations are available [6, 7, 12, 18]. From these results, we can easily infer that (assume $r > 0$ in the following)

$$E_r(x,y) = \max(0, 1 - \tfrac{1}{r}|x-y|)$$

is a $T_{\mathbf{L}}$-equivalence on $\mathbb{R}$, where $T_{\mathbf{L}}(x,y) = \max(0, x+y-1)$ denotes the Łukasiewicz t-norm. Analogously,

$$E'_r(x,y) = \exp(-\tfrac{1}{r}|x-y|)$$

is a $T_{\mathbf{P}}$-equivalence on $\mathbb{R}$, where $T_{\mathbf{P}}(x,y) = xy$ denotes the product t-norm.

Based on a general representation theorem for strongly complete fuzzy orderings [4], we can further prove that

$$L_r(x,y) = \min(1, \max(0, 1 - \tfrac{1}{r}(x-y)))$$

is a strongly complete $T_{\mathbf{L}}$-E_r-ordering on $\mathbb{R}$ and that

$$L'_r(x,y) = \min(1, \exp(-\tfrac{1}{r}(x-y))$$

is a strongly complete $T_{\mathbf{P}}$-E'_r-ordering on $\mathbb{R}$. As $T_{\mathbf{L}} \leq T_{\mathbf{P}}$, we can trivially conclude that L'_r is also a strongly complete $T_{\mathbf{L}}$-E'_r-ordering.

In order to generalize the notion of concordant and discordant pairs, we need the notion of a strict fuzzy ordering. We call a binary fuzzy relation R a *strict fuzzy ordering* with respect to T and a T-equivalence E, for brevity *strict T-E-ordering*, if it is irreflexive (i.e. $R(x,x) = 0$ for all $x \in X$), T-transitive, and E-extensional, that is,

$$T(E(x,x'), E(y,y'), R(x,y)) \leq R(x',y')$$

for all $x, x', y, y', z \in X$ [5].

Given a T-E-ordering L,

$$R(x,y) = \min(L(x,y), N_T(L(y,x))), \tag{2}$$

where $N_T(x) = \sup\{y \in [0,1] \mid T(x,y) = 0\}$ is the residual negation of T, is the most appropriate choice for extracting a strict fuzzy ordering from a given fuzzy ordering L (for a detailed argumentation, see [5]). From this construction, we can infer that the fuzzy relation

$$R_r(x,y) = \min(1, \max(0, \tfrac{1}{r}(y-x)))$$

is a strict $T_{\mathbf{L}}$-E_r-ordering and that

$$R'_r(x,y) = \max(0, 1 - \exp(-\tfrac{1}{r}(y-x)))$$

is a strict $T_{\mathbf{L}}$-E'_r-ordering.

If a given $T_{\mathbf{L}}$-E-ordering L is strongly complete, it can be proved that the fuzzy relation R defined as in (2) simplifies to

$$R(x,y) = 1 - L(y,x)$$

and that the following holds:

$$R(x,y) + E(x,y) + R(y,x) = 1 \tag{3}$$

$$\min(R(x,y), R(y,x)) = 0 \tag{4}$$

5 A Fuzzy Ordering-Based Rank Correlation Coefficient

The previous section has provided us with the apparatus that is necessary to define a generalized rank correlation measure. Assume that the data are given as in (1) again (with $x_i \in X$ and $y_i \in Y$ for all $i = 1, \ldots, n$). Further assume that we are given two $T_{\mathbf{L}}$-equivalences $E_X : X^2 \to [0,1]$ and $E_Y : Y^2 \to [0,1]$, a strongly complete $T_{\mathbf{L}}$-E_X-ordering $L_X : X^2 \to [0,1]$ and a strongly complete $T_{\mathbf{L}}$-E_Y-ordering $L_Y : Y^2 \to [0,1]$. Therefore, we can define a strict $T_{\mathbf{L}}$-E_X-ordering on X as $R_X(x_1,x_2) = 1 - L_X(x_2,x_1)$ and a strict $T_{\mathbf{L}}$-E_Y-ordering on Y as $R_Y(y_1,y_2) = 1 - L_Y(y_2,y_1)$.

Spearman's rho is based on rankings. Rankings are crisp concepts in which it is not easy to accommodate degrees of relationship in a straightforward way. Thus it is more meaningful to use pairwise comparisons to

define a concept of rank correlation, just like Kendall's tau and the gamma measure do.

Given an index pair (i, j), we can compute the degree to which $((x_i, y_i), (x_j, y_j))$ is a concordant pair as

$$\tilde{C}(i,j) = \min(R_X(x_i, x_j), R_Y(y_i, y_j))$$

and the degree to which $((x_i, y_i), (x_j, y_j))$ is a discordant pair as

$$\tilde{D}(i,j) = \min(R_X(x_i, x_j), R_Y(y_j, y_i)).$$

If we adopt the simple sigma count idea to measure the cardinality of a fuzzy set [8], we can compute the numbers of concordant pairs $\tilde{C}$ and discordant pairs $\tilde{D}$, respectively, as

$$\tilde{C} = \sum_{i=1}^{n} \sum_{j \neq i} \tilde{C}(i,j),$$
$$\tilde{D} = \sum_{i=1}^{n} \sum_{j \neq i} \tilde{D}(i,j).$$

The question arises whether we should attempt to generalize τ_a, τ_b or γ. As the main motivation is to get rid of the influence of close-to-ties pairs in the presence of noise, it is immediate that the idea behind γ is the most promising one. So, with the assumptions from above, we define our *fuzzy ordering-based rank correlation measure* $\tilde{\gamma}$ as

$$\tilde{\gamma} = \frac{\tilde{C} - \tilde{D}}{\tilde{C} + \tilde{D}}.$$

To interpret the meaning of $\tilde{\gamma}$, we note that, for all index pairs (i, j), the equality

$$\tilde{C}(i,j) + \tilde{C}(j,i) + \tilde{D}(i,j) + \tilde{D}(j,i) + \tilde{T}(i,j) = 1 \quad (5)$$

holds, where $\tilde{T}(i, j)$ denotes the degree to which (i, j) is a tie in either variable:

$$\tilde{T}(i,j) = \max(E_X(x_i, x_j), E_Y(y_i, y_j))$$

Moreover, we can infer the following:

$$\tilde{C} = \sum_{i=1}^{n} \sum_{j>i} (\tilde{C}(i,j) + \tilde{C}(j,i))$$
$$\tilde{D} = \sum_{i=1}^{n} \sum_{j>i} (\tilde{D}(i,j) + \tilde{D}(i,j))$$

Thus, by (5), $\tilde{C} + \tilde{D}$ equals the number of non-tie pairs if we consider each choice of indices i, j only once (in contrast to considering (i, j) *and* (j, i) independently for each i and j). So $\tilde{\gamma}$ measures the difference of concordant and discordant pairs relative to the number of non-tie pairs; the concept of "tiedness" is a fuzzy one, however.

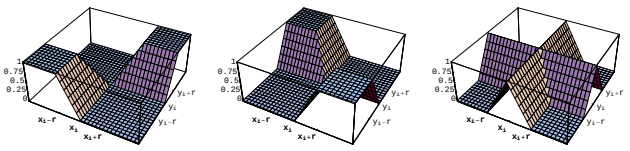


Figure 2: $\tilde{C}(i,j) + \tilde{C}(j,i)$ (left), $\tilde{D}(i,j) + \tilde{D}(j,i)$ (middle), and $\tilde{T}(i,j)$ (right) plotted as functions of x_j and y_j for fixed x_i and y_i (using the relations E_r and R_r).

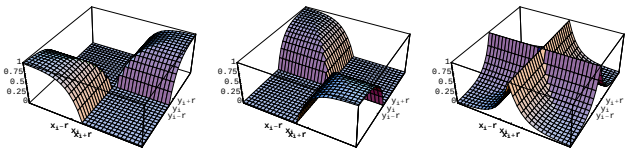


Figure 3: $\tilde{C}(i,j) + \tilde{C}(j,i)$ (left), $\tilde{D}(i,j) + \tilde{D}(j,i)$ (middle), and $\tilde{T}(i,j)$ (right) plotted as functions of x_j and y_j for fixed x_i and y_i (using the relations E'_r and R'_r).

It is obvious that, in case that E_X and E_Y are crisp equalities and that R_X and R_Y are crisp linear strict orderings, that $\tilde{\gamma}$ coincides with γ. So what is the difference if R_X and R_Y are non-trivial fuzzy relations? The above interpretation shows that concordant/discordant pairs are counted more if they are dissimilar and less if they are similar—which perfectly corresponds to our intention. Let us demonstrate this fact with an example.

Assume $X = Y = \mathbb{R}$, $E_X = E_Y = E_r$, and $R_X = R_Y = R_r$ for some $r > 0$. Fixing some x_i and y_i and considering $\tilde{C}(i,j) + \tilde{C}(j,i)$, $\tilde{D}(i,j) + \tilde{D}(j,i)$, and $\tilde{T}(i,j)$ as functions of the two variables x_j and y_j, the graphs shown in Figure 2 can be obtained. It can be seen that pairs are counted fully if $|x_i - x_j| > r$ and $|y_i - y_j| > r$ (i.e. like in the classical γ measure). If one of the two distances is smaller than r, the pair is considered as a tie to the corresponding degree $\tilde{T}(i,j)$ and only counted to a degree of $1 - \tilde{T}(i,j)$. One also sees that, if r is chosen so large that $|x_i - x_j| \leq r$ and $|y_i - y_j| \leq r$ for all pairs, all pairs are counted to a degree proportionally to the minimum of these two distances. If the relations $E_X = E_Y = E'_r$, and $R_X = R_Y = R'_r$ are used, the effect is qualitatively similar, r also controls to which degree a close-to-tie pair is counted, also in a monotonic, yet asymptotic fashion (see Figure 3).

It is clear from the above examples that, the smaller r, the more $\tilde{\gamma}$ resembles to γ. For both, the variant based on E_r/R_r and the variant based on E'_r/R'_r, it can be proved that $\tilde{\gamma}$ converges to γ for $r \to 0$.

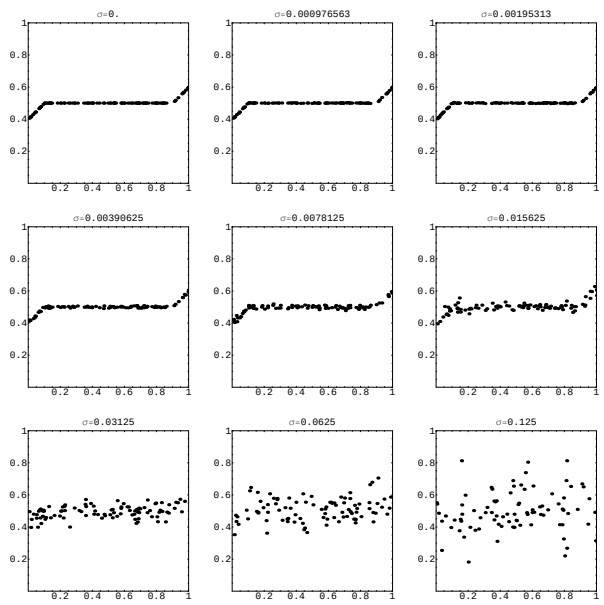


Figure 4: Different data sets obtained from contaminating a non-decreasing relationship by normally distributed noise with different standard deviations.

Another property of $\tilde{\gamma}$ is immediate to see: if the fuzzy relations R_X and R_Y are continuous (assuming that this notion makes sense on X and Y), then $\tilde{\gamma}$ depends continuously on the data set $(x_i, y_i)_{i=1}^n$.

6 Experiments

Let us first reconsider the example from Section 3. More specifically, we are given 100 uniformly distributed random values $(x_1, \ldots, x_{100})$ from the unit interval. The list $(y_1, \ldots, y_{100})$ is computed as $y_i = f(x_i)$, where f is a simple, piecewise linear, non-decreasing function that has a relatively large flat area. In order to study how different rank correlation measures react to noise, we contaminated the data points with additive, independent, normally distributed noise with 0 mean and standard deviation σ. Figure 4 shows these data sets. Figure 6 displays the results that we obtained for different rank correlation measures. We compared ρ, τ_b, γ and different variants of $\tilde{\gamma}$. Every line in Figure 6 corresponds to the results obtained by one rank correlation measure depending on the noise level σ. The two lines for τ_b (dotted, black) and γ (dotted, light gray) coincide except for no noise ($\sigma = 0$). Both lines reveal that these two measures react to noise in an non-robust way. More or less the same is true for ρ (dotted, medium gray). The other lines correspond to different variants of $\tilde{\gamma}$. Solid lines correspond to $\tilde{\gamma}$ using R_r and dashed lines denote the results for $\tilde{\gamma}$ using R'_r (where we use the same r for both components). We used $r = 0.05$ (black), $r = 0.2$ (medium gray), and $r = 0.5$ (light gray). We see that all six different variants react to the noise in a more robust way than the three crisp measures. Clearly, the higher r, the more noise is neglected. Note, however, that, the larger r, the more difficult it is for $\tilde{\gamma}$ to find out whether there are slightly non-monotonic parts in the data.

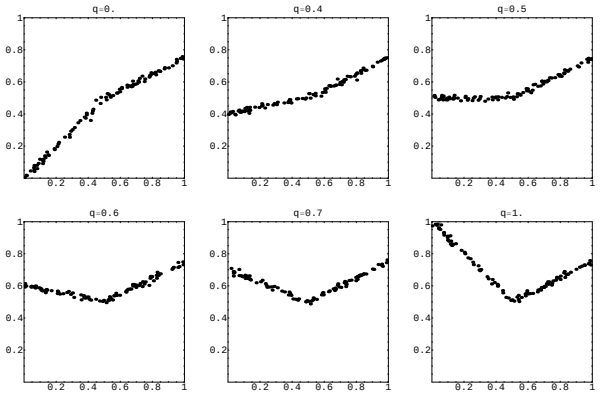


Figure 5: Noisy data sets that correspond to monotonic ($q \leq 0.5$) and non-monotonic relationships ($q > 0.5$).

So let us consider a different setting. Now we fix the noise level $\sigma = 0.01$ and use different functions to create the second list $(y_1, \ldots, y_{100})$. Right of $x = 0.5$, we use $f(x) = \frac{x}{2} + \frac{1}{4}$ and to the left or $x = 0.5$, we linearly interpolate between $(0, q)$ and $(0.5, 0.5)$. It is clear, that this relationship is monotonic if and only if $q \leq 0.5$. The data sets are displayed in Figure 5 and the results are presented in Figure 7, where we use the same conventions to distinguish the lines as in Figure 6. We see that all variants of $\tilde{\gamma}$ show acceptable results for $q \leq 0.5$, whereas ρ, τ_b and γ again have problems to handle the noise in case of the large proportion of ties that occurs for $q = 0.5$. We also see that $\tilde{\gamma}$ already yields significantly lower values for $q = 0.6$ in the case $r = 0.05$ (no matter which of the two variants is considered). For larger r, however, we see that $\tilde{\gamma}$ cannot detect the slight non-monotonicity for $q = 0.6$ that well. These two examples demonstrate that, when choosing r, there is a trade-off between robustness (the larger r, the better) and sensitivity (the smaller r, the better).

As a third set of experiments, we have tried to figure out the variance of $\tilde{\gamma}$. For this study, we have computed all rank correlation measures used in the above experiments for different test data several times and computed the variance of the results. In all experiments, we have encountered that τ_b and γ had higher variances than all variants of $\tilde{\gamma}$. The variances we obtained for different variants of $\tilde{\gamma}$ obeyed a simple

and unsurprising rule: the larger r, the smaller the variance. Interestingly, the variances we obtained for Spearman's ρ were also very low, comparable to the lower values for $\tilde{\gamma}$ with a large r.

Note that the authors have carried out numerous experiments to solidify the above claims. As the space in this paper is limited, we just quoted the most interesting and demonstrative results.

7 Concluding Remarks

This paper, as the appellative term "towards" in the title suggests, attempts to present first ideas that the authors consider promising. The examples of the previous section are intended to support this viewpoint. They are illustrative and indicative, but they cannot replace a formal investigation of the properties of $\tilde{\gamma}$. As it has been done exhaustively for Spearman's rho and Kendall's tau, a significance analysis and a variance analysis have to be carried out. Note, however, that this cannot be done analogously for $\tilde{\gamma}$. Both Spearman's rho and Kendall's tau are fully determined by the ranking of the lists $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$. Thus, combinatorial techniques can be used to study variances and significance levels [11]—not so for $\tilde{\gamma}$ that always depends on the distance relationships of the values, too, so this analysis can only be done by some distribution assumptions. These studies are left to future research.

To determine the right choice for the parameter r is another open question. As we have noted above, there is a trade-off between robustness on the one side and sensitivity/significance on the other side. So this topic goes hand in hand with a more formal statistical analysis. Profound results concerning the choice of r, again, can only be expected with specific distribution assumptions. In any case, we want to note in advance that $\tilde{\gamma}$ depends continuously on r, so at least we can be sure that $\tilde{\gamma}$ will react robust to slightly sub-optimal choices of r.

Finally, we would like to remark that this investigation was inspired by a problem in bioinformatics: how to infer sets of co-transcribed genes in procaryotic genomes (so-called *operons*) from the gene expression levels measured by microarray experiments [3, 15, **?**]. It will also be subject of future research to evaluate the rank correlation measures introduced in this paper in this domain.

References

[1] H. Abdi. Coefficients of correlation, alienation and determination. In N. J. Salkind, editor, *Encyclopedia of Measurement and Statistics*. Sage, Thousand Oaks, CA, 2007.

[2] H. Abdi. The Kendall rank correlation coefficient. In N. J. Salkind, editor, *Encyclopedia of Measurement and Statistics*. Sage, Thousand Oaks, CA, 2007.

[3] J. Bockhorst, Y. Qiu, J. Glasner, M. Liu, F. Blattner, and M. Craven. Predicting bacterial transcription units using sequence and expression data. *Bioinformatics*, 19(Suppl. 1):i34–i43, 2003.

[4] U. Bodenhofer. A similarity-based generalization of fuzzy orderings preserving the classical axioms. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 8(5):593–610, 2000.

[5] U. Bodenhofer and M. Demirci. Strict fuzzy orderings in a similarity-based setting. In *Proc. Joint 4th Conf. of the European Society for Fuzzy Logic and Technology and 11 Recontres Francophones sur la Logique Floue et ses Applications*, pages 297–302, Barcelona, September 2005.

[6] B. De Baets and R. Mesiar. Pseudo-metrics and T-equivalences. *J. Fuzzy Math.*, 5(2):471–481, 1997.

[7] B. De Baets and R. Mesiar. Metrics and T-equalities. *J. Math. Anal. Appl.*, 267:331–347, 2002.

[8] A. DeLuca and S. Termini. A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Inf. Control*, 20:301–312, 1972.

[9] L. A. Goodman and W. H. Kruskal. Measures of association for cross classifications. *J. Amer. Statist. Assoc.*, 49(268):732–764, 1954.

[10] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–93, 1938.

[11] M. G. Kendall. *Rank Correlation Methods*. Charles Griffin & Co., London, third edition, 1962.

[12] F. Klawonn. Fuzzy sets and vague environments. *Fuzzy Sets and Systems*, 66:207–221, 1994.

[13] W. H. Kruskal. Ordinal measures of association. *J. Amer. Statist. Assoc.*, 53(284):814–861, 1958.

[14] K. Pearson. Notes on the history of correlation. *Biometrika*, 13:25–45, 1920.

[15] C. Sabatti, L. Rohlin, M.-K. Oh, and J. C. Liao. Co-expression pattern from DNA microarray experiments as a tool for operon prediction. *Nucleic Acids Res.*, 30(13):2886–2893, 2002.

[16] C. Spearman. The proof and measurement of association between two things. *Am. J. Psychol.*, 15(1):72–101, 1904.

[17] C. Spearman. Demonstration of formulae for true measurement of correlation. *Am. J. Psychol.*, 18(2):161–169, 1907.

[18] E. Trillas and L. Valverde. An inquiry into indistinguishability operators. In H. J. Skala, S. Termini, and E. Trillas, editors, *Aspects of Vagueness*, pages 231–256. Reidel, Dordrecht, 1984.

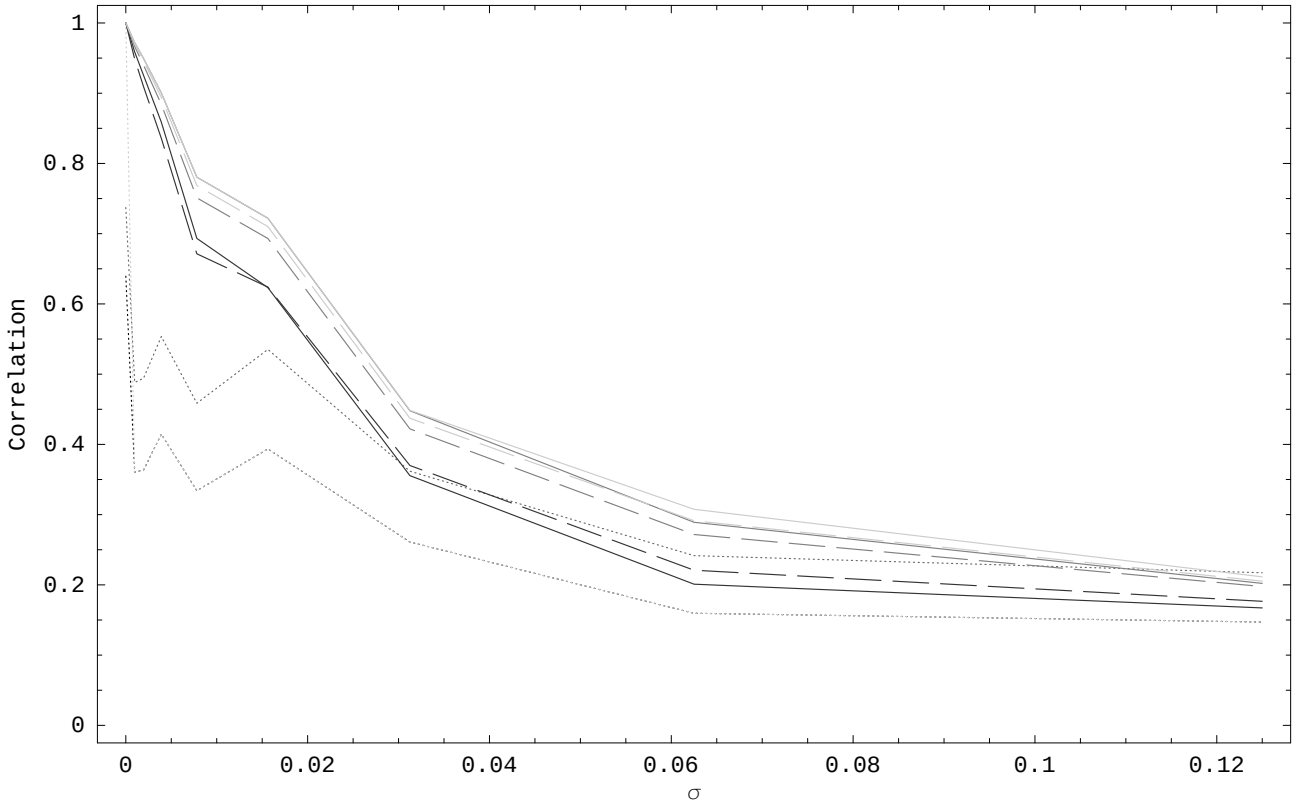


Figure 6: Results obtained by applying different rank correlation measures to the data sets shown in Figure 4.

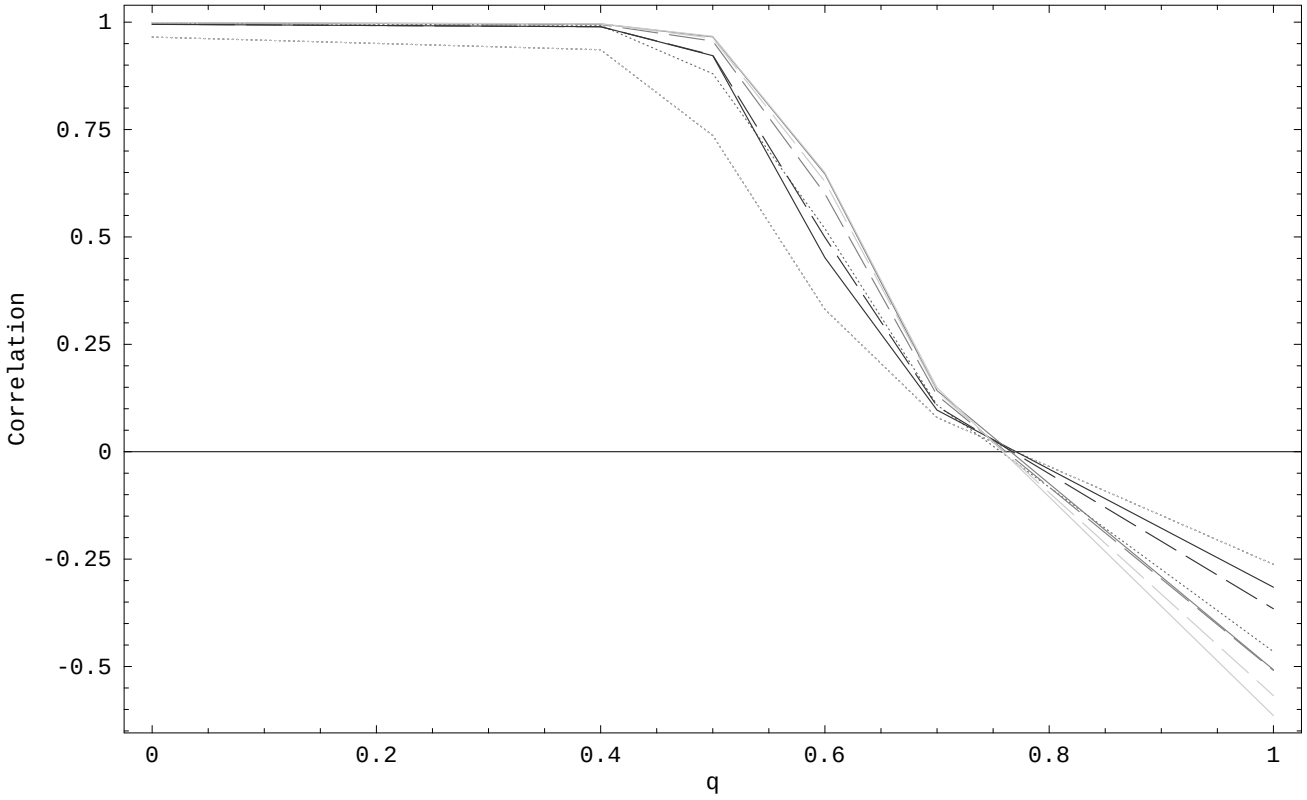


Figure 7: Results obtained by applying different rank correlation measures to the data sets shown in Figure 5.

Learning of Decision Trees with Incremental ID3

Zheng He 0656422
e-mail: hz_23@hotmail.com

May 4, 2007

Abstract

Nowadays, classification plays a very important role in data mining problems, and has been studied comprehensively in some research communities. During the recent decades, classification has also been successfully applied to various fields such as marketing management, credit approval, customer segment, medical diagnosis, performance prediction and Shopping options. The scientists developed several classification models like Bayesian classification, neural networks, statistical models such as linear or quadratic discriminants, genetic models and decision trees. Decision trees are one of the most popular techniques of data mining among these models.

A decision tree is a model of a decision procedure for discriminating the class of given instances. Every node of the decision tree determines either a class or a specific test which partitions the space of instances. In these tree structures, a leaf node is a node containing a class name and a non-leaf node is a node that contains an attribute test with a branch to another decision tree for each possible value of the attribute. One of the most famous and valuable algorithms for building an optimal decision tree is **ID3**, which served as a basis for plenty of variations and improvements.

This kind of methods have a good performance in batch mode(off-line) problems. However, sometimes, the data set involves millions of records or grows as a stream. Non-incremental algorithms could hardly operate in this situation, while incremental ones provide an effective tool to deal with the problems in a step-by-step way. Based on **ID3** algorithm, **ID4** and **ID5R** were designed to learn decision trees incrementally, and under certain circumstances, they can build the same decision tree as their off-line version **ID3** algorithm.

RANSAC – model estimation algorithm

presenter: Matěj Šmíd
Software competence center Hagenberg
matej.smid@scch.at

Abstract

Finding mathematical model in data where are many outliers can be difficult problem. RANSAC is simple but robust solution for this kind of problems. Algorithm was developed in 1981 and its usability was proven on many real world examples.

Introduction

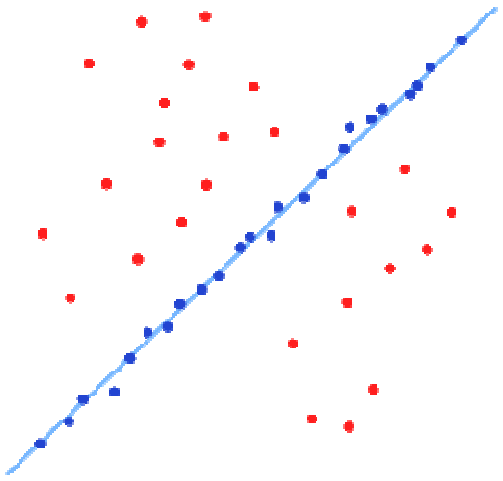
RANSAC is an abbreviation for "RANdom SAmple Consensus". It is an algorithm to estimate parameters of a mathematical model from a set of observed data which contains outliers. The algorithm was first published by Fischler and Bolles in 1981.

A basic assumption is that the data consists of "inliers", i. e., data points which can be explained by some set of model parameters, and "outliers" which are data points that do not fit the model. In addition to this, the data points can be subject to noise. The outliers can come, e. g., from extreme values of the noise or from erroneous measurements or incorrect hypotheses about the interpretation of data. RANSAC also assumes that, given a (usually small) set of inliers, there exists a procedure which can estimate the parameters of a model that optimally explains or fits this data.

Example

A simple example is fitting of a 2D line to set of observations. Assuming that this set contains both inliers, i.e., points which approximately can be fitted to a line, and outliers, points which cannot be fitted to this line, a simple least squares method for line fitting will in general produce a line with a bad fit to the inliers. The reason

1 Data with many outliers, line has to be fitted.



2 Fitted line with RANSAC, outliers have no influence on result.

is that it is optimally fitted to all points, including the outliers. RANSAC, on the other hand, can produce a model which is only computed from the inliers, provided that the probability of choosing only inliers in the selection of data points is sufficiently high. There is no guarantee for this situation, however, and there are a number of algorithm parameters which must be carefully chosen to keep the level of probability reasonably high.

Overview

The input to the RANSAC algorithm is a set of observed data values, a parameterized model which can explain or be fitted to the observations, and some confidence parameters.

RANSAC achieves its goal by iteratively selecting a random subset of the original data points. These points are hypothetical inliers and this hypothesis is then tested as follows. A model is fitted to the hypothetical inliers, that is, all free parameters of the model are reconstructed from the point set. All other data points are then tested against the fitted model, that is, for every point of the remaining set, the algorithm determines how well the point fits to the estimated model. If it fits well, that point is also considered as a hypothetical inlier. If sufficiently many points have been classified as hypothetical inliers relative to the estimated model, then we have a model which is reasonably good. However, it has only been estimated from the initial set of hypothetical inliers, so we reestimate the model from the entire set of point's hypothetical inliers. At the same time, we also estimate the error of the inliers relative to the model.

This procedure is then repeated a fixed number of times, each time producing either a model which is rejected because too few points are classified as inliers or a refined model together with a corresponding error measure. In the latter case, we keep the refined model if its error is lower than the last saved model.

Algorithm

The generic RANSAC algorithm works as follows:

```
input:
    data – a set of observed data points
    model – a model that can be fitted to data points
    n – the minimum number of data values required to fit the model
    k – the maximum number of iterations allowed in the algorithm
    t – a threshold value for determining when a data point fits a
        model
    d – the number of close data values required to assert that a
        model fits well to data
output:
    bestfit – model parameters which best fit the data (or nil if no
    good model is found)

iterations := 0
bestfit := nil
besterr := infinity
while iterations < k
    maybeinliers := n randomly selected values from data
    maybemodel := model parameters fitted to maybeinliers
```

```
    alsoinliers := empty set

    for every point in data not in maybeinliers
        if point fits maybemodel with an error smaller than t
             add point to alsoinliers

    if the number of elements in alsoinliers is > d
        (this implies that we may have found a good model now test
        how good it is)
        bettermodel := model parameters fitted to all points in
                       maybeinliers and alsoinliers
        thiserr := a measure of how well model fits these points
        if thiserr < besterr
            bestfit := bettermodel
            besterr := thiserr

    increment iterations

return bestfit
```

Advantages and disadvantages

An advantage of RANSAC is its ability to do robust estimation of the model parameters, i.e., it can estimate the parameters with a high degree of accuracy even when outliers are present in the data set. A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters. If an upper time bound is used, the solution obtained may not be the most optimal one.

Applications

The RANSAC algorithm is often used in computer vision, e.g., to simultaneously solve the correspondence problem and estimate the fundamental matrix related to a pair of stereo cameras.

References

- M. A. Fischler and R. C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. of the ACM* **24**: 381--395. DOI:10.1145/358669.358692.
- David A. Forsyth and Jean Ponce (2003). *Computer Vision, a modern approach*. Prentice Hall. ISBN ISBN 0-13-085198-1.
- Richard Hartley and Andrew Zisserman (2003). *Multiple View Geometry in computer vision*, 2nd edition, Cambridge University Press.

Newton Methods for Nonlinear Inverse Problems with Random Noise

Frank Bauer[1], Thorsten Hohage[2] and Axel Munk[2]
[1]: FLLL
[2]: University of Göttingen
frank.bauer@jku.at, hohage@math.uni-goettingen.de, munk@math.uni-goettingen.de

---

The solution of nonlinear operator equations $F(a) = u$ by iterative regularization methods with deterministic noise has been studied intensively over the last decade. However, we are not aware of any convergence and convergence rate results of iterative regularization methods for nonlinear inverse problems with random noise. Moreover well-known stopping rules like Morozov's discrepancy principle do not work in this situation.

We will present a slightly modified iteratively regularized Gauss-Newton method with the Lepskij-type balancing principle as a-posteriori stopping rule.

We do not need any restrictive assumptions on the structure or the degree of nonlinearity of the operator F, but only have to require Lipschitz continuity of F' and a sufficiently strong source condition. As a result we obtain almost optimal rates of convergence over a range of Hölder smoothness classes.

HDFormGen: A Fast Nonlinear Approximation Formula Generator for Very High Dimensional Data Based on Variable Selection and Genetic Programming

Werner Groißböck

Department of Knowledge-Based Mathematical Systems
Fuzzy Logic Laboratorium Linz-Hagenberg
Johannes Kepler University Linz, A-4040 Linz, Austria
werner.groissboeck@jku.at

Summary. A new approach for finding nonlinear approximation formulas for very high-dimensional data is presented. This method has been developed for static data analysis, but it can be used for dynamic data analysis as well. The method is based on linear regression, but instead of the original variables we use nonlinear terms with these variables. Such a formula is still linear in the parameters, so ordinary least squares methods can be applied to find the globally optimal parameters. We use an accelerated version of genetic programming to find the optimal nonlinear terms, and we use variable selection methods to select those terms leading to an approximation formula which shows an optimal balance of accuracy and simplicity. In general, evolutionary methods like genetic programming tend to produce many individuals with low fitness. To save computation time, an early stopping strategy in case of low fitness is used. The method was tested with three benchmark data sets (the auto-mpg data set and the CPU data set in the UCI repository http://www.ics.uci.edu/ mlearn/MLRepository.html and the friedman data set in the KEEL repository http://sci2s.ugr.es/keel/). Although these data sets are only low dimensional and thus not in the core application area of our method, for the auto-mpg data set, an approximation formula has been determined, whose accuracy is comparable to the benchmark papers, for the CPU data set, an approximation formula has been achieved which is more exact than most of the benchmark papers, and for the friedman data set, an approximation formula has been determined which is more exact than all of the benchmark papers found so far.

1 Introduction

In the car industry, an engine test bench system is used which can measure up to 1500 variables. From time to time, some parts of the measurement system are in an invalid state, maybe because one of the sensors is overheated. To

safe time and money, such an invalid state has to be detected as soon as possible, and the experiment has to be aborted as soon as possible. So a system is needed, which can detect faults.
For most of the variables measured useful expert knowledge is not available. For this reason, only data driven methods can be used. Different methods are available. The major challenge is that the methods have to deal with a very high dimensionality.

The method HDFormGen (A fast nonlinear **For**mula **Gen**erator for **H**igh **D**imensional Data) can be used to find a nonlinear approximation formula for very high dimensional data. To demonstrate the strength of our approach, the following artificial data set with 201 variables and 800 entries has been constructed: The variables $x1, x2, ..., x200$ are filled with independent standard normally distributed numbers. The remaining variable (which we call y) is determined with the following formula:

$$\begin{aligned} y =& x1 \cdot (0.3 \cdot x5 - 0.6 \cdot (x3 \cdot x5 + x2 \cdot x6) \\ &+ 0.2 \cdot (x2 \cdot x4 \cdot x6 + x2 \cdot x3 \cdot x7 + x3 \cdot x4 \cdot x5 - x5 \cdot x6 \cdot x7)) \end{aligned} \tag{1}$$

We want to find an approximation formula for the variable y. [1] So we want to see if our only data driven method can find any reasonable results. After running our algorithm for half an hour (all our results have been processed on a 1600MHz pentium laptop) the following formula has been achieved:

$$\begin{aligned} y =& 9.4589e-008 \\ &- 0.6 \cdot (x6 \cdot (x2 \cdot x1)) \\ &- 0.6 \cdot ((x3 \cdot x5) \cdot x1) \\ &+ 0.3 \cdot (x1 \cdot x5) \\ &+ 0.2 \cdot ((x7 \cdot x3) \cdot (x2 \cdot x1)) \\ &+ 0.2 \cdot (((x1 \cdot x4) \cdot x5) \cdot x3) \\ &+ 0.2 \cdot ((x6 \cdot x1) \cdot (x4 \cdot x2)) \\ &- 0.2 \cdot ((x1 \cdot (x5 \cdot x6)) \cdot x7) \end{aligned} \tag{2}$$

This formula is nearly identical to a simplified form of the formula in 1. The only difference is the constant $9.4589e-008$, which is caused by the limitations of machine accuracy. The most important question is: Does the algorithm still work, when data sets containing noise have to be analyzed? To answer this question, the data set described above is used again, but now to each variable a certain amount of noise is added, before our algorithm is applied. As noise we use independent standard normally distributed numbers, which are divided by ten.

[1] The estimated standard deviation of y is 0.81222, so it is not zero, which would make the task trivial.

After an average time consumption of about 4.5 hours, the following approximation formula (for the noisy data set) can be achieved:

$$\begin{aligned} y =& 0.0097468 \\ & - 0.56631 \cdot (x1 \cdot (x2 \cdot x6)) \\ & - 0.57815 \cdot ((x1 \cdot x3) \cdot x5) \\ & + 0.28516 \cdot (x1 \cdot x5) \\ & + 0.18876 \cdot (x2 \cdot ((x1 \cdot x7) \cdot x3)) \\ & + 0.18276 \cdot (x1 \cdot (x5 \cdot (x3 \cdot x4))) \\ & - 0.17787 \cdot (x1 \cdot (x5 \cdot (x6 \cdot x7))) \\ & + 0.18482 \cdot (((x4 \cdot x6) \cdot x1) \cdot x2) \end{aligned} \tag{3}$$

This formula is not identical to the formulas above. But if the subterms are compared, then we can see that all the subterms in formula 3 can also be found in formula 2 and vice versa. So the only real differences are the exact values of the parameters before each nonlinear subterm in the formulas. For example for the subterm with $x1, x3$ and $x5$, we get the parameter -0.57815 instead of the parameter -0.6. This slight modification of the parameters is a consequence of the noise that has been added to the data variables. If a data based method is used, and if you have to deal with noisy data, then a certain amount of error in the models achieved can never be avoided.
Conclusion: We have been able to find a formula that is 'nearly' equivalent to the formulas in 1 and 2. The only relevant differences are the real parameters in the formulas. For finding the correct parameters, we use the least squares algorithm, which finds the globally optimal parameter setting. Finally, the correct structure of the formula is found, and the globally optimal parameter setting!

2 The approximation formula generator HDFormGen

In this paper, the new algorithm *HDFormGen* (A **For**mula **Gen**erator for **H**igh **D**imensional Data) is introduced which is able to find an approximation formula with nonlinear terms for a high dimensional regression data set. With this algorithm, formulas similar to the following can been achieved:

$$y = \beta_0 + \beta_1 \cdot x1 \cdot x100 + \beta_2 \cdot \sin(x77) + \beta_3 \cdot \exp(x5/x6)$$

The basic idea of the algorithm is the following:

- The structure of each of the nonlinear terms in the whole formula is found and optimized with the use of genetic programming (see [5]).
- The parameters of the formula can be optimized easily with a least squares algorithm. This can only be done, if the formula is linear in the parameters, so the genetic programming tool must not generate terms which contain additional parameters.

There is another aspect that has to be considered:
The terms that are used in the approximation formula finally shall be as uncorrelated to each other as possible. We want an approximation formula which is on the one hand as simple as possible, and on the other hand as exact as possible. So we have to find the most important nonlinear terms, such that the regression formula based on these terms is as good as possible. Variable selection methods like *forward selection* have been designed to fulfill this task. In *HDFormGen* a variant of forward selection is used. For this reason, the basic concept of *forward selection* will be explained in the following rows:

- At first, the most important variable (or nonlinear term) is selected. This is that variable (or term) which is correlated strongest to the actual dependent y.
- Then the effects of the variables/terms selected so far are subtracted from the original dependent y. This is necessary to avoid that variables that are highly correlated to the first choice will be chosen again and again.
- Then, again the most important variable/term is selected.
- And again, the dependent is modified, such that the effects of the variables/terms chosen already are eliminated.
- Continue in this manner, until enough variables/terms are selected.

3 The new algorithm HDFormGen

3.1 The core of the new algorithm

In the following, the original dependent is called y. At the beginning, the actual dependent is the original dependent $y_{actual} = y$. Later y_{actual} will be modified. The constant term $c = (1, \ldots, 1)^T$ is always the first variable that is chosen. But this variable is not counted as real variable. The algorithm performs the following steps:

1. An accelerated version of genetic programming (including a population of individuals and a crossover operator) is used to generate millions of very simple formulas. We select that formula x_A which is best correlated with the actual dependent y_{actual}. We look only at the absolute value of the correlation coefficient.
2. Then we modify y_{actual} such that all the parts of y that can be approximated with the regressors already chosen are subtracted, setting y_{actual} to $y - \hat{y}(c, x_A)$. Here $\hat{y}(c, x_A)$ is the linear best approximation of y with the use of the regressors c and x_A. We can say, y_{actual} is y made orthogonal to the regressors already chosen.
3. Once again the accelerated version of genetic programming is used to generate millions of very simple formulas. And now we select that formula x_B which is correlated strongest with the actual dependent y_{actual}. We look only at absolute values again.

4. Then once again, y_{actual} is made orthogonal to the regressors already chosen, so we set y_{actual} to $y - \hat{y}(c, x_A, x_B)$.
5. Continue in this manner, until a given number of regressor terms is selected or some other termination criterion is fulfilled.

3.2 The accelerated version of genetic programming - an overview

Stopping the calculation of the correlation coefficient as early as possible, when it can be seen that the checked individual is not worth spending additional time, accelerates the algorithm enormously. But how can this be carried out, if we have a population of individuals and not a single individual? In the following lines the major steps of the accelerated genetic programming algorithm are described.

1. Generate an initial population with *popsize* individuals.
2. Evaluate each individual for $n1$ points of the training data set and estimate the correlation coefficient with the actual dependent by using only these $n1$ points.
3. Determine the $popsize_{small}$ best correlated individuals out of *popsize*, based on the estimated correlation coefficient. We look only at the absolute value of the correlation coefficient.
4. For these $popsize_{small}$ chosen individuals the *e*xact value of the fitness function (i.e. the absolute value of the correlation coefficient) using *a*ll the points of the training data set has to be calculated.
5. Produce a new generation of *popsize* out of the $popsize_{small}$ chosen individuals:
   - Repeat the following, until we have enough new individuals. Choose randomly two of the $popsize_{small}$ individuals and compare their fitness. The better one is called the winner, and the other one is called the loser. Let the winner produce two offsprings, one is an exact copy of the winner, and the other offspring is made via crossover (as crossover partner, one of the $popsize_{small}$ individuals is chosen, which is neither the winner nor the loser).
   - The individual which is the best so far is always copied into the next generation ('elitism').
   - A small part of the new generation is produced in the same way as the initial population. This is one way of avoiding the problem with local optima. A mutation is not needed any more.
6. Go to step 2, until a termination criterion is fulfilled.

- As termination criterion, we usually take that a specific number of generations is reached.
- The parameter *popsize* determines, how many individuals are evolved in the genetic programming algorithm. The parameter *popsize* can take any positive integer number. The larger *popsize* is, the more computation time

is needed, and the better the results are. In our experiments, a *popsize* of 5000 has been used successfully.
- The parameter $n1$ tells the algorithm, how many points are used to get a quick estimation of the correlation coefficient. $n1$ can be an arbitrary positive integer, but $n1$ shall not exceed the number of training data points. In our experiments, settings of $n1 = 30$, $n1 = 50$ and $n1 = 100$ have been used successfully.
- The parameter $popsize_{small}$ determines, how many individuals of the total population are selected to be examined in detail. The value of $popsize_{small}$ shall be much smaller than *popsize*, for example $popsize/10$.

4 Variants of the Formula Generator Algorithm Applied To Standard Benchmark Data Sets

4.1 The data set cpu

The data set 'cpu' can be found in the directory 'cpu-performance' of the UCI-repository, which can be found in the following address:

`http://www.ics.uci.edu/~mlearn/MLRepository.html`

Number of instances: 209
Number of attributes: 10
The data set contains the following attributes:

vendor name: string
model name: string
MYCT: machine cycle time in nanoseconds (integer)
MMIN: minimum main memory in kilobytes (integer)
MMAX: maximum main memory in kilobytes (integer)
CACH: cache memory in kilobytes (integer)
CHMIN: minimum channels in units (integer)
CHMAX: maximum channels in units (integer)
PRP: published relative performance (integer); the dependent variable;
ERP: estimated relative performance via linear regression (integer)

At first we deleted the attributes *vendorname* and *modelname* because our algorithm can not handle strings. Furthermore the data set contains the attribute ERP, which is an old estimation for PRP. So we have to delete the attribute ERP, because we do not want to generate an approximation formula by using the results of an old approximation. This would be too easy. So finally we have only 7 attributes remaining. Before the core of our algorithm has been run, we split the data into two parts: `70%` of the 209 instances have been randomly chosen to play the role of the training data. And the other `30%` play the role of the test-data.

Our algorithm has been started 10 times. Roughly 3.7 seconds are necessary per term for performing the evolutionary part of the algorithm. Totally we received ten approximation formulas, with an average MAE of 23.33 determined for the test data set. The worst MAE is only 25.15, and the best MAE is 23.06. The best formula is the following:

$$\begin{aligned} PRP =& 16.344 \\ &+ 0.0032443 \cdot (sqrtabs((MMIN \cdot (MMAX \cdot CHMAX)))) \\ &+ 0.7936 \cdot ((CACH - CHMAX) - \sin(CHMAX)) \end{aligned} \tag{4}$$

The MSE of this formula is 1394.9, and the RMSE is 37.348. In our standard benchmark paper (see [7]), various different methods have been tried out. The best method leads to an MAE of 38.0. So compared to this paper, our method leads to a more exact approximation.

Additionally, newer papers (see [10], [12], [2] and [1]) have been found, where the data set cpu is used.

Conclusion: In these papers, totally 30 variants of standard methods have been tried out. Only in 5 out of 30 cases, our approximation formula is outperformed.

4.2 The data set friedman

The data set 'friedman' can be found in the KEEL repository, in the following location:

http://sci2s.ugr.es/keel/datasets1.php?SID&codeds=36

In the keel repository, benchmark papers can be found. For the friedman data set, a quite actual (2004) benchmark paper is mentioned via the abbreviation 'Lee04' (see [6]).

We try to design our experiments as similar as possible to the benchmark paper, to get comparable results. In the benchmark paper, the following is done:

'This is a synthetic benchmark data set. Each sample consists of five inputs and one output. The formula for the data generation is $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4^4 + 5x_5^5 + \varepsilon$, where ε is a Gaussian random noise $N(0, 1)$, and $x_1, ..., x_5$ are uniformly distributed over the domain $[0, 1]$. 1400 samples were created, of which 200 samples were randomly chosen for network training and 200 samples for validation. The remaining 1000 samples were used for network testing.'

In the KEEL repository, the data sets are already available as described in [6]. So we have a 200 sample training data set, and a 200 sample validation data set, and a 1000 sample test data set. Unlike the benchmark paper, we do

not need any validation data. So we only take the 200 sample training data set to find an approximation formula, and we take the 1000 sample test data set to determine the quality. As a quality measure, here the MSE is used, according to the benchmark data.

Our formula detection algorithm has been run 20 times. Here we need 13 seconds for each term, and 30 second for finding the total formula, because the formula consists of two nonlinear terms, and four seconds are needed in the non-evolutionary part of the algorithm. The best formula that we get is the following:

$$\begin{aligned} out =& 4.8843 \\ &+ 10.1761 \cdot (in4 + \sin((in2 \cdot (in1 + (in1 + in1))))) \\ &- 5.3183 \cdot (\sin((in3 + (in3 + in3))) - in5) \end{aligned} \tag{5}$$

The MAE of this formula is 0.889281, the MSE of this formula is 1.23629, and the RMSE is 1.11189. In the benchmark-paper (see [6]), the best method leads to an MSE of 4.502. So our formula is much more exact.

Cross-validation and the data set friedman

For the dataset friedman, a tenfold cross validation experiment has been performed. For this experiment, a 1200-sample version of the friedman data set has been used, which can be downloaded from the KEEL repository, in the following location:

http://sci2s.ugr.es/keel/datasets1.php?SID&codeds=36

After the cross-validation, we have to calculate the average error measures on the test data files. We get an average MAE of 0.8394133, an average MSE of 1.1127881, and an average $RMSE$ of 1.0537323 .

So with cross validation, we finally get ten formulas. The formula, which reaches the best quality on the corresponding test data set, is the following formula:

$$\begin{aligned} out =& 4.9946 \\ &- 10.1215 \cdot (\sin(((in2 \cdot in1) \cdot (1.051813 \cdot -2.92026))) - in4) \\ &+ 20.4701 \cdot ((in3 \cdot in3) - ((-0.2465477 \cdot in5) + in3)) \\ &+ 2.9015 \cdot ((0.3611782 - (in1 \cdot in2)) \cdot (in1 \cdot \sin(in3))) \end{aligned} \tag{6}$$

This formula reached (on the test data set number 10) an MAE of 0.786382, an MSE of 0.963263, and an RMSE of 0.981459627 . The name of the corresponding test data file in the KEEL repository is 'Friedman-10-10tst.dat', so everybody is invited to check the quality of the formula! It has to be mentioned that here the best formula out of ten has been selected (via the

test data), so we can not expected to get such a result in average. The average qualities have been stated above, and are more important.

Conclusion: For the friedman data set, all the benchmark papers that we found so far (see [3], [6] and [11]), have been outperformed by our method.

References

1. M. Birattari, G. Bontempi and H. Bersini, "Lazy Learning Meets the Recursive Least Squares Algorithm", MIT Press, Advances in Neural Information Processing Systems 11, Cambridge, 1999.
2. M. Ceci, A. Appice and D. Malerba, "Comparing Simplification Methods for Model Trees with Regression and Splitting Nodes", Springer Lecture Notes in Computer Science, Volume 2871/2003, ISBN: 3-540-20256-0, 2003.
3. Q. Fu, S. X. Hu, S. Y. Zhao, "Clustering-based selective neural network ensemble", Journal of Zhejiang University SCIENCE 2005 6A(5), ISSN 1009-3095, doi:10.1631/jzus.2005.A0387, 2005.
4. Hastie, T., Tibshirani, R., and Friedman J. , "The Elements of Statistical Learning: Data Mining, Inference, and Prediction.", Springer Berlin, 2001.
5. J. R. Koza, "Genetic Programming", The MIT Press, Cambridge, Massachusetts, 1992.
6. W. M. Lee, C. P. Lim, K. K. Yuen, S. M. Lo, "A Hybrid Neural Network Model for Noisy Data Regression", IEEE Transactions on Systems, Man and Cybernetics, Part B 34:2, Pages 951-960, 2004.
7. Merz, C. J., Pazzani, M. J., "Combining Neural Network Regression Estimates with Regularized Linear Weights", Advances in Neural Information Processing Systems 9, edited by M.C. Mozer, M.I. Jordan, and T. Petsche, 1997.
8. A. Miller, "Subset Selection in Regression - Second Edition", ISBN 1-58488-171-2 Chapman & Hall/CRC Boca Raton London New York Washington, D.C. 2002.
9. O. Nelles, "Nonlinear System Identification - From Classical Approaches to Neural Networks and Fuzzy Models", ISBN 3-540-67369-5 Springer-Verlag Berlin Heidelberg New York, 2001.
10. N. Rooney, D. W. Patterson, S. S. Anand and A. Tsymbal, "Random subspacing for regression ensembles", The Florida AI Research Society Conference, 2004.
11. D. P. Solomatine, M. B. L. A. Siek, "Flexible and Optimal M5 Model Trees with Applications to Flow Predictions", 6^{th} International Conference on Hydroinformatics, Liong, Phoon & Babovic (eds), ISBN 981-238-787-0 , World Scientific Publishing Company, 2004.
12. R. Setiono, W. K. Leow and J. Y. L. Thong, "Opening the neural network black box: an algorithm for extracting rules from function approximating artificial neural networks", Proceedings of the twenty first international conference on Information systems, Australia, 2000.