International Journal for Uncertainty Quantification, 2 (1): 73–94 (2012)

ON THE ROLE OF DATA MINING TECHNIQUES IN UNCERTAINTY QUANTIFICATION

Chandrika Kamath*

Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, California, 94551, USA

Original Manuscript Submitted: 9/6/2011; Final Draft Received: 12/7/2011

Techniques from scientific data mining are increasingly being used to analyze and understand data from scientific observations, simulations, and experiments. These methods provide scientists the opportunity to automate the tedious manual processing of the data, control complex systems, and gain insights into the phenomenon being modeled or observed. This process of data-driven scientific inference borrows ideas and solutions from a range of fields including machine learning, image and video processing, statistics, high-performance computing, and pattern recognition. The tasks involved in these analyses include the extraction of structures from the data, the identification of representative features for these structures, dimension reduction, and building predictive and descriptive models. At first glance, data mining and data-driven analysis may appear unrelated to stochastic modeling and uncertainty quantification. But, as we show in this paper, there are commonalities in the problems addressed and techniques used, providing the two communities the opportunity to benefit from the expertise and experiences of each other.

KEY WORDS: *classification, machine learning, principal component analysis, high-dimensional methods, data mining, uncertainty quantification*

1. INTRODUCTION

Data mining is the semi-automated discovery of patterns, associations, anomalies, and statistically significant structures in data. Over the last decade, as data mining techniques have matured and evolved to address new types of data and analysis problems, they have been successfully applied to a wide range of problems. Many of the more commonly known examples of data mining occur in business and commercial applications, such as credit-card fraud detection, recommender systems in on-line shopping, as well as text and web mining in search engines. However, data mining techniques also are being applied in various science domains, where they are used to automate tedious manual analysis, understand and control complex systems, and gain insights into the phenomenon being modeled or observed [1].

In this paper, using example problems, we describe the techniques that are used in scientific data mining. We start with an overview of the end-to-end process of data mining as applied to data sets from scientific applications. We then discuss in more detail the algorithms commonly used in the steps that comprise the data mining process. Since data mining borrows ideas and solutions from a range of fields including machine learning, image and video processing, statistics, high-performance computing, and pattern recognition, there are a vast number of techniques used in each one of these steps. While an in-depth discussion of all these techniques is out of the scope of this paper, we provide references for those interested in additional information.

Before we describe the data mining process in detail, it might be helpful to consider why the work being done in the data mining and related communities may be relevant to stochastic modeling and uncertainty quantification. First, the tasks addressed in data mining are very similar to those performed in stochastic modeling and uncertainty quantification. These include building surrogate models, addressing issues arising from high dimensionality of data

*Correspond to Chandrika Kamath, E-mail: kamath2@llnl.gov, URL: http://ckamath.org

2152–5080/12/$35.00

sets, extracting useful information from massive data sets, and designing computational and physical experiments to generate the data. This gives an opportunity for the two communities to benefit from each other, even though, in some cases, the motivation for the techniques developed may be different. Second, the topic of data plays an important role in uncertainty quantification as ensembles of simulations are run to characterize uncertainty in the results. These simulations generate a large volume of data that must be compared to theory, experiments, and other simulations. Techniques from data mining are, therefore, likely to be relevant in both the analysis of the data from the simulations and in identifying the next set of simulations to run in the ensemble. Finally, there has been much work done in the data analysis community, especially in statistics and machine learning, on methods for reasoning in the presence of uncertainty. Solutions using Bayesian techniques [2] and probabilistic graphical models [3, 4] are increasingly being considered for practical problems. The data mining community also is becoming aware of the need to incoporate uncertainty into their algorithms, given their increasing use in decision support and for the analysis of data with uncertain or missing values. As a result, there is a wealth of solutions in data mining techniques that could be exploited for use in stochastic modeling and uncertainty quantification.

This paper is written from the perspective of a data mining practitioner. The techniques selected for a task are those we have found useful in addressing the practical problems we have encountered in our work. Several general texts have been included in the references for more details on other techniques that might be relevant. Our intent is to introduce the stochastic modeling and uncertainty quantification communities to techniques that may be relevant in the context of their problems, although these techniques were originally developed in a different domain. We also hope that the inclusion of this paper in a special issue focusing on stochastic modeling and uncertainty quantification would encourage data miners to borrow and adapt ideas from these communities and apply them to data analysis problems.

2. THE SCIENTIFIC DATA MINING PROCESS

At a high level, any data mining endeavor, whether in the context of commercial or scientific data, can be considered as composed of two main steps—the pre-processing of the data and the identification of the patterns in the data. In the case of scientific applications, the data from observations, simulations, and experiments are often in the form of multi-variate time series, structured and unstructured spatio-temporal data, or images. However, we are interested in patterns among the objects in these data; for example, patterns among galaxies found in images from observational astronomy. So, we first need to identify the objects in the data and find suitable representations for them. The raw data may also be of poor quality, with noise, missing values, and low contrast. These data are frequently multivariate, and may have been obtained from different sensors or simulations at different resolutions. To convert these data into a form suitable for pattern recognition requires a substantial amount of pre-processing. In our experience analyzing data from several scientific domains, we have found that an iterative and interactive approach composed of multiple steps, as outlined in Fig. 1, has worked well. We next briefly describe the tasks in these steps [1]; these tasks, and the algorithms for them, are further discussed in Sections 3–8, along with references for the interested reader.

In problems where the data sets are very large, we may want to make the initial analysis tractable by working with a smaller sample (say, every nth time step in a simulation) or use multi-resolution techniques to reduce the size of the data being processed. In other problems, where the data are from different sources, we may need to fuse them so we can exploit the complementary information in these sources. This may include converting time series data from different sensors that are sampled at different frequencies into data sampled at the same rate. Or, we may first need to register images taken of phenomena over time before we can track objects of interest in them.

Next, we may want to improve the quality of the data though techniques such as signal and image de-noising and image contrast enhancement. This is especially true for data collected during experiments and observations. Following this cleanup step, we may need to identify objects in the data for problems where we are interested in patterns among the structures or objects found in the data. These structures could be galaxies in astronomy images, coherent structures in simulations of fluid turbulence, or fragments in experimental images used for understanding material fragmentation.

Once the structures in the data have been identified, we need to represent them using low-level "features." These features are any extractable measurement or attribute from the data, and should not be confused with the term "features" often used in some domains to describe objects of interest in the data, such as vortices in fluid flow. In data

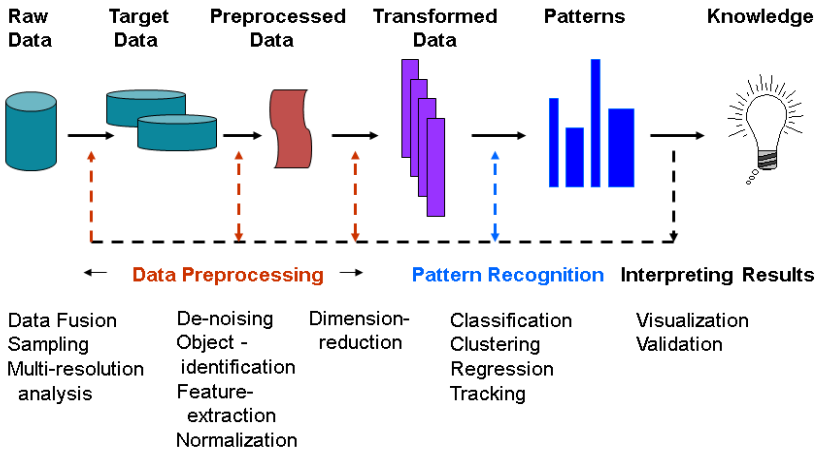

FIG. 1: The end-to-end process of scientific data mining.

mining, the term feature represents characteristics of the objects that can be used to identify the patterns; for example, the angle between two blobs that form a galaxy, the input parameters for a simulation, or the integrated value of a variable over all grid points that comprise a structure in a simulation. Once the features for each object in the data have been extracted, we may need to normalize them and check them for correctness.

In some problems, we may have a large number of features representing each object as we may extract far more features than necessary, not knowing which ones are the most relevant and discriminating. The next step in the data mining process is dimension reduction, where we reduce the number of features (the dimension of the problem) by identifying the important ones.

Finally, we are ready to identify the patterns in the data. Depending on the task, this can be done using techniques such as classification, clustering, anomaly detection, association rules, and so on. These patterns are evaluated by both the data miners and the domain scientists, and the process refined until satisfactory results are obtained.

We next make several general observations on the process of scientific data mining. The data flow diagram presented in Fig. 1 is one that we have found to cover the needs of the scientific applications we have encountered. However, variations and enhancements may be necessary as required by an application. For example, the order in which the tasks are done may change from application to application and some tasks may be skipped if not required. While much of the focus tends to be on the task of finding the patterns in the data, it is the data pre-processing steps that are far more time consuming, frequently taking up to 80-90% of the total time for analysis. However, as we shall see, it is critical that these steps be performed correctly for any data mining endeavor to be successful.

The scientific data mining process is very interactive, with the domain scientists involved in every step, starting with an initial formulation of the problem to providing information on the data collection process, verifying the objects extracted from the data, identifying representative features for the objects, and most importantly, validating the results obtained at each step. The process is also an iterative process. The results of any one step may indicate that a previous step needs to be refined. For example, the identification of patterns may indicate some features that are key to discrimination are not rotation invariant and, therefore, objects that are rotated versions of objects in the training set are not being labeled correctly. Or, the error rate of the pattern recognition step could be high, indicating that the features extracted are not representative enough of the patterns being considered or that the quality of the training data could be improved. As a result of these issues, data mining software can seldom be used as black boxes. Instead, scientific data mining is a careful and considered application of techniques in close collaboration with the domain scientists.

Each of the tasks in Fig. 1 can be implemented using several algorithms. These algorithms differ in their suitability for a problem, the assumptions they make about the data, their computational complexity, the accuracy of the results,

their robustness to input parameters, and their interpretability. Often, several algorithms may need to be tried before one suitable for the data and problem is found. While many of the techniques used in data mining are independent of the problem and application, there are times when we may need to design algorithms that are tuned to the characteristics of the data or problem. This is especially true for the tasks of object identification and feature extraction, and also may be relevant in the analysis of massive data sets, where we could exploit problem-specific characteristics to devise a computationally inexpensive solution.

Finally, we observe that data mining borrows and builds on techniques from several disciplines ranging from statistics to machine learning, pattern recognition, and signal and image processing. As a multi-disciplinary field, it lies at the intersection of applied mathematics, computer science, and applications.

In the rest of this paper, I consider the tasks in the data mining process in more detail, and using examples, illustrate some of the algorithms that can be used to address them. These tasks include the pre-processing of the data to improve their quality; the identification of objects of interest in the data; the extraction of features to represent the objects; the reduction in the number of features, that is, the dimension of the problem; the building of models from the data; and the generation of the data themselves.

3. PRE-PROCESSING THE DATA TO IMPROVE QUALITY

Science data, especially those obtained from experiments and observations, are often of poor quality, with noise, missing values or other characteristics that can make further processing difficult. These data must, therefore, be "cleaned" first. This can be done using simple approaches from traditional image processing [5, 6], such as the application of a mean or a Gaussian filter to reduce the noise, or using histogram equalization to improve the contrast. More sophisticated methods, such as diffusion techniques based on partial differential equations [7, 8], as well as statistical techniques [9, 10], are also an option.

As an example, consider the images in Fig. 2. These are part of a problem to understand the Richtmyer-Meshkov instability [11] that results when an impulsive acceleration is applied to the interface separating two fluids of different densities, for example, as a result of a shock wave striking the interface perpendicularly. Such instabilities arise in diverse situations such as supernovas, oceans, and supersonic combustion, and are, therefore, the subject of much research. To understand what happens to the fluids over time, scientists have been using computer simulations to model the instability. Our analysis task in this problem was code validation, that is, we wanted to compare the simulations with experiments. The image in the left panel of Fig. 2 is obtained using planar laser-induced fluorescence (PLIF) imagery in an experimental setup where the interface between a column of acetone/air mixture on the top of a shock tube and a column of sulphur hexachloride at the bottom is perturbed by a shock wave at Mach 1.3 [12].

To compare the experiments with the simulations, we considered an approach that first identified the mushroom-shaped structures in the data, extracted characteristics for these structures, such as the height and width of the mushrooms, and then performed the comparison using these characteristics [13]. However, extracting the mushroom-shaped structures is difficult as the noise in the image (in the form of vertical lines) is as strong as the signal, especially in the low-contrast regions of the image. To reduce the noise, we borrowed an idea from restoration of archival films,

FIG. 2: Reducing domain-specific noise in images from experiments of Richtmyer-Meshkov instability. Left: original image; right: after reducing the noise that appears as vertical lines in the image.

which are often corrupted by line scratches. These images can be restored using model-based statistical approaches, which first detect the lines and then reconstruct the data that have been corrupted [14, 15]. In our approach, we first segmented the image to identify the edges using a Canny edge detector [5, 6], and then focused on the vertical lines, performing a median filtering only on the noise segments, resulting in the image in the right panel of Fig. 2. In other words, we essentially used a model of the noise, which appears as vertical lines, to perform local noise reduction.

As another example, consider the images in Fig. 3. On the left is a subset of an original image taken of a material as it fragments. The lighter areas are the fragments of the material, while the darker regions represent the space or gaps between the fragments. The goal of the analysis is to obtain statistics both for the fragments (such as their size) and the gaps (such as their length and width). The distributions of these characteristics, in the form of histograms, are used to provide a concise summary of each image. As we need to process several images of varying quality, we are interested in techniques that can be automated, require few parameters, and are relatively insensitive to the values of these parameters. A key challenge in the analysis is the rather large variation in intensity from the upper-left of the image to the lower-right corner. In fact, the intensity in the darker gap regions at the upper-left corner is close to the intensity of the lighter fragments at the lower-right corner. Second, some fragments, such as the ones near the bottom of the image, have non-uniform intensity, with lighter pixels in one part and darker pixels in the other. This makes it challenging to identify the material fragments in the image.

Our approach to solving this problem was to start by addressing the varying illumination in the image. Non-uniform illumination can be considered a form of multiplicative noise, a problem commonly addressed using the Retinex algorithm.This method, first proposed by Land [16], represents the intensity at each pixel in the image as the product of the reflectance and the illuminance at that pixel. By taking the natural logarithm of the image, we can "subtract out" the non-uniformity of the illumination. In our work, we used the multi-scale Retinex technique of Jobson et al. [17]. Let $I(x, y)$ represent the two-dimensional image. Then, the single-scale retinex output, $R(x, y)$, is defined as [18]

$$R(x, y) = \log_e[(I(x, y)] - \log_e[F(x, y) * I(x, y)],$$

where the second term represents the illumination and is the convolution of the image with a Gaussian filter F of the form

$$F(x, y) = K \exp[-(x^2 + y^2)/\sigma^2],$$

where

$$K = \iint F(x, y)\, dx\, dy = 1,$$

and the standard deviation, σ, of the Gaussian determines the scale. The multi-scale Retinex method is a weighted sum of N applications of the single-scale algorithm

$$MSR(x, y) = \sum_{i=1}^{N} w_i R_i(x, y),$$

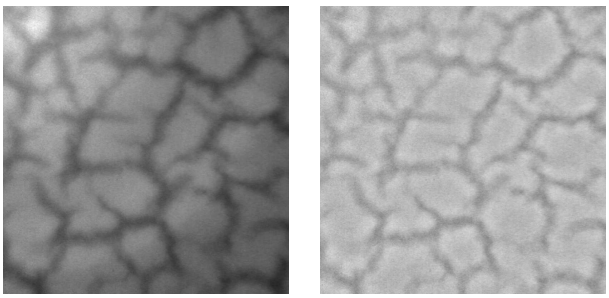

FIG. 3: Reducing the variation in intensity in images of fragmentation of materials. Left: original image; right: after application of the multi-scale Retinex algorithm.

where each $R_i(x, y)$ is obtained using a different scale σ_i and the weights sum to 1. For our images, we used two equally weighted scales, one small with $\sigma = 20$, and the other larger, with $\sigma = 80$. The right panel in Fig. 3 shows how the application of the multi-scale Retinex algorithm makes the illumination much more uniform across the image. We further enhance the image using smoothing by a Gaussian filter and a minimum mean-squared error filter that reduces the noise in the image prior to the identification of the fragments [19].

4. IDENTIFYING THE OBJECTS OF INTEREST IN THE DATA

In some problems, identifying the objects of interest is relatively easy. For example, if we are analyzing the relationships between the inputs and outputs of an ensemble of simulations, then each simulation is an "object." However, when we want to extract statistics on fragments in an image or compare a simulation to an experiment, we first need to identify the fragments or the structures in the simulation and experimental data that could be used for the comparison. There are three broad categories of techniques we can use to identify the structures of interest in science data. We can focus on the boundary of the structure, we can focus on its interior, or we can use domain specific methods. We next describe these approaches briefly.

In problems where there is a large gradient at the boundary of the objects, it can be exploited to separate the object from the background. For example, in Fig. 2, the mushroom-shaped objects are in black, while the background is in gray. Thus, simple filters that calculate the gradient in the x and y directions can be applied at each pixel in the image. A simple thresholding on the magnitude of the gradient would then identify the high-contrast pixels in the image, which would be at the boundary, or the "edge," of the objects [6]. However, the results are likely to be dependent on the threshold—too low a value will identify many pixels as edge pixels while too high a value will result in gaps in the edges. A more robust approach to edge detection is that proposed by Canny [20]; once the gradient at each pixel has been determined, we apply a "thinning" process using non-maximal suppression, where a pixel is retained only if its gradient is a local maximum in the direction perpendicular to the edge. Next, instead of a simple thresholding, we use thresholding with hysteresis. All pixels with a gradient magnitude below a low threshold are dropped, while those above a high threshold are retained. Pixels with gradient magnitude in between the two thresholds are retained only if they can be recursively connected to a pixel above the high threshold. The effect of hysteresis thresholding is to close gaps in the edges. Figure 4 shows the edges found using the Canny method for the problem of comparison of experimental and simulation data for the Richtmyer-Meshov instability. The figures in the middle panel show that if the experimental images are not de-noised (as described in Section 3), we have a large number of spurious edges that meet the thresholding criterion.

Other commonly used edge detection techniques include the smallest univalue segment assimilating nucleas (SUSAN) approach [21], which considers a circular region centered at each pixel and evaluates the similarity of the center pixel to the the rest of pixels in the region, as well as level sets, which evolve a suitably defined partial differential equation [22].

While many edge detection techniques can be used successfully when there is a sharp gradient between the objects of interest and the background, they often perform poorly in practice when the image has regions of low contrast, resulting in edges with gaps or missed edges. In such cases, an alternative approach is to focus on the interior of the objects and exploit the fact that the points in the interior tend to be similar to each other. Such region growing methods come in many flavors [6]. In Fig. 5, we describe the use of region growing methods in two and three dimensions in the context of the analysis of the Rayleigh-Taylor instability. This occurs when an initially perturbed interface between a heavier fluid, which is on top of a lighter fluid, is allowed to grow under the influence of gravity. Finger-like structures of the lighter fluid penetrate the heavier fluid in what are referred to as "bubbles," while "spikes" of heavier fluid move into the lighter fluid. Panel (a) in Fig. 5 shows the simulation at early time, where the heavier fluid (in red) is on top of the lighter fluid (in blue). With time, these structures, which are initially distinct, continue to evolve. They may grow, change shape, split, merge with surrounding structures, or shrink in size relative to other structures that grow and overtake them. We analyzed data from two three-dimensional simulations: one, a 30 terabyte large-eddy simulation (LES) and the other an 80 terabyte direct numerical simulation (DNS) [23, 24]. The analysis task was to identify, count, and track the bubble and spike structures in the simulations with the goal of understanding the dynamics of these structures.

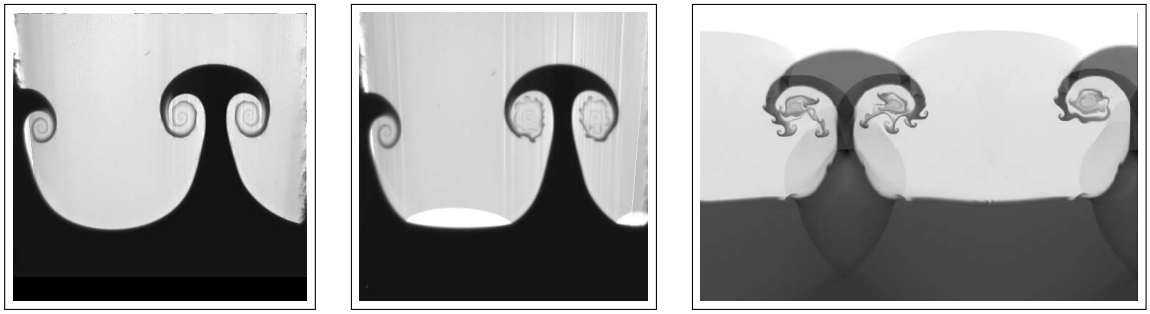

FIG. 4: Using the Canny edge detector to identify the mushroom-shaped structures in experimental images of Richtmyer-Meshkov instability: (a) de-noised experimental image at early time; (b) noisy original image at mid-time; (c) simulation output at mid-time; (d)–(f) the edges found in the images in the top row.

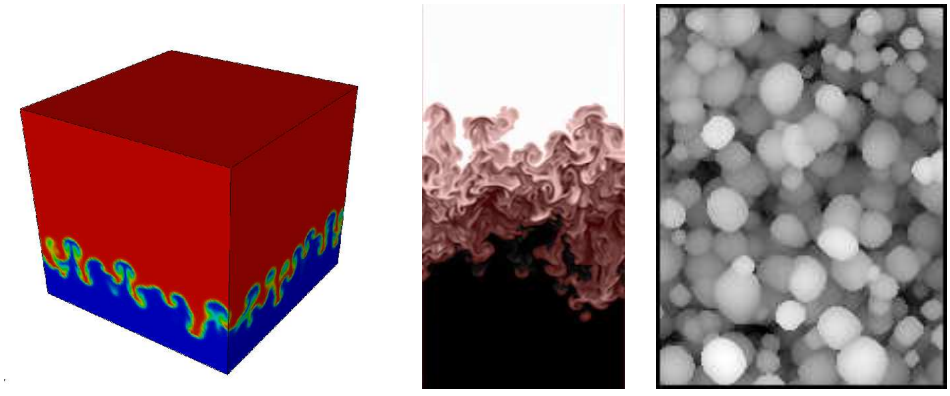

FIG. 5: Identifying the bubble and spike structures in three-dimensional simulations of Rayleigh-Taylor instability. (a) A cube showing the three-dimensional domain with bubbles of the lighter fluid (in blue) penetrating the heavier fluid (in red) while spikes of the heavier fluid enter the lighter fluid. (b) A two-dimensional slice of the LES data, showing the region left over (in pink) after the application of a three-dimensional region growing algorithm from the top and bottom. (c) The top view of a column of the three-dimensional data, showing the bubble boundary. The image shows the height of the bubble surface from the initial fluid interface; a brighter bubble is closer to the top of the cube.

As described in [25–28], there were several challenges to the analysis of these data sets, ranging from the massive size and distributed nature of the data, to the lack of a clear definition for the bubble and spike structures. Our solution

approach was to first identify the bubble and spike boundaries by exploiting the density variable. Since the top fluid has density 3 and the lower fluid has density 1, we can grow a region each from the top and the bottom of the cube to identify the "background" or the volume of unmixed fluid. The region in the middle represents the mixed fluid and its boundary in three dimensions is the boundary of the bubble and spike structures. We grow the top region downward by adding pixels that are connected to the region and whose neighbors meet a high threshold criterion. Similarly, we grow the bottom region upward by adding pixels that are connected to the region and whose neighbors meet a low threshold criterion. At some point, when they no longer meet the threshold criteria, both the top and bottom regions stop growing; this identifies the surface of the bubble and spike structures, a two-dimensional slice of which is shown in Fig. 5, panel (b), where the region in pink is what remains after the top and bottom regions stop growing.

Once the three-dimensional surfaces of the bubble and spike structures were found, we transformed the data into two dimensions by considering the height (depth) maps that are essentially the top (bottom) view of the bubble height (spike depth), relative to the original fluid interface. Figure 5, panel (c), shows the height of the bubble surface from the initial fluid interface; a brighter bubble is closer to the top of the cube. We can then identify the bubbles and spikes in these two-dimensional images by applying the region-growing method to these height-depth maps. Our goal is to group pixels that are close to each other and of a similar height. So, we start with the highest pixel, and grow a region around it by adding neighboring pixels that are close to the highest pixel in their intensity value (which is the height of the bubble surface). When the region cannot grow any further, as no neighboring pixels satisfy the height constraint, we select the highest pixel among those remaining and grow a region around it, and so on, until all pixels have been assigned a region. The resulting regions, with their boundaries identified in red, are shown in Fig. 6, left panel, for the DNS calculation. Since the image is obviously over-segmented, we can clean it by removing small regions and merging those completely contained within another. This results in the cleaned image shown in the right panel. From this, it is easy to count the number of bubbles and spikes at any time step.

Although one can apply traditional image analysis techniques in a straightforward manner to identify objects in the data, sometimes it helps to exploit any special properties the data may have to devise a more efficient and effective algorithm. For example, in counting the bubbles and spikes in the Rayleigh-Taylor simulations, we found that we could exploit the values of the x and y velocities at the three-dimensional bubble and spike boundaries to devise a fast algorithm for counting the tips of the bubbles and spikes [25, 28]. As another example of exploiting domain information, consider the two images in Fig. 7, which are examples of bent-double galaxies from the Faint Images of the Radio Sky at Twenty Centimeters (FIRST) survey [29]. Our analysis goal was to build a predictive model that could differentiate between radio-emitting galaxies that had a bent-double morphology and those that did not [30]. The astronomers had processed the data collected by the telescopes to create images of the radio sky.

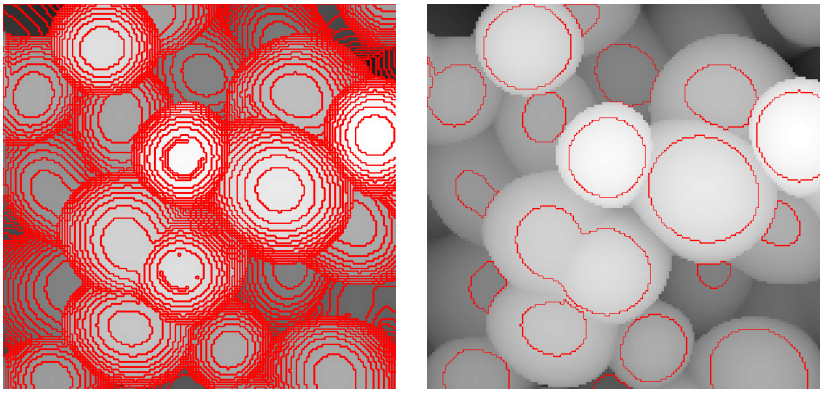

FIG. 6: Identifying the bubble and spike boundaries in two dimensions by applying the region growing technique. Left: the regions identified in a height map of the DNS calculation. The value of the pixel intensities, representing the height of the bubble surface from the initial fluid interface, is similar for each region, but differs across regions. Right: the same image after cleanup to merge regions that are contained within one another and to remove small regions.

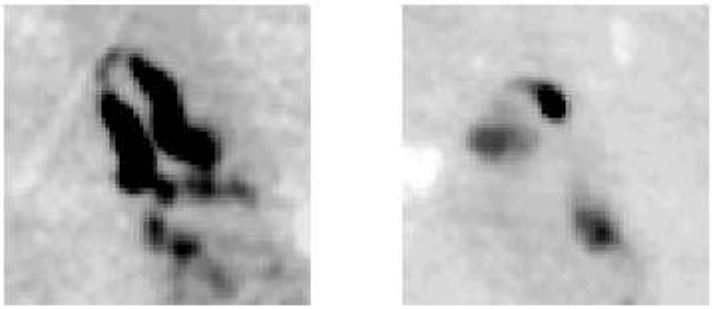

FIG. 7: Images from the Faint Images of the Radio Sky at Twenty centimeters (FIRST) survey, showing two galaxies with bent-double morphology. The galaxies in the left and the right panels are composed of 11 and three entries in the catalog, respectively, indicating the number of elliptic Gaussians required for accurate representation.

A large part of these images was just background noise, with occasional regions with the galaxies. To identify the galaxies, the astronomers exploited the fact that they were composed of "blobs," which could be approximated well by two-dimensional elliptic Gaussians. So, they created a catalog, each entry of which was the information on each of these Gaussians, such as the peak flux, the major and minor axes, the coordinates of the center, and the position angle of the major axis measured in degrees counter-clockwise from North. A reconstruction of the galaxies using the information in the elliptic Gaussians indicated that the catalog data were a good representation of the original images.

As another example of a domain-specific approach to identifying objects of interest in the data, consider the data shown in Fig. 8 generated as part of a three-dimensional simulation to understand turbulence in burning plasma in tokamaks [31]. Panel (a) in Fig. 8 is the ion heat flux variable in a two-dimensional poloidal plane of the tokamak. There are clearly visible structures in the data and our task was to identify these structures and extract statistics for them. Since the structures are radially aligned, we explored the values of the ion heat flux on the grid points that lie on a circle centered at the center of the "hole" (which is the magnetic axis of the tokamak). These grid points form a flux surface. Panel (b) of Fig. 8 shows the values of the ion heat flux (in blue) on one of the flux surfaces. This indicates that if we consider the valleys in the curve, and drop the grid points that occur at each valley, we should be able to disconnect the structures on this flux surface. A threshold suitably selected by applying this idea to several flux surfaces worked well to isolate the structures at early time, as shown in panel (c). However, we found that the ion heat flux variable was quite noisy at the later time steps and the identification of the valleys became more difficult. A solution to this problem was found when we noticed that another variable, the electrostatic potential [shown in red in panel (b)], was far better behaved at late time and had its peaks and valleys coincident with the valleys of the ion heat flux. Thus, we were able to exploit domain information to extract the structures of interest in a robust manner for all time steps.

And, finally, we would be remiss if we did not mention model-based approaches to identify objects in data. Here, we exploit a model of the objects of interest—for example, circular or rectangular objects—to extract them from the data. We can also build models in the form of distributions. For example, a common technique for calculating a threshold is to consider the problem as a two class problem and classify a grid point or pixel as belonging to one class or the other by modeling the intensities as a mixture of two Gaussians. If the two classes are well separated, there is a distinct valley in the resulting bi-modal distribution that can be selected as the threshold. However, as shown in the left panel in Fig. 9, this valley can be difficult to determine for our problem on extracting statistics for images of material fragmentation; there is no real valley discernible in the histogram of the image. However, we clearly see an asymmetry in the histogram, and a threshold selected at the point of asymmetry on the left side of the histogram works well in identifying the fragments in the image (as shown in the right panel). This threshold is also close to that obtained using Otsu's method [32], which considers the threshold as one that maximizes the between-class variance derived using the histogram of the image intensity. Note that the thresholding leaves behind some stray isolated pixels, or small groups of pixels, which must be removed by post-processing the segmented image [19].

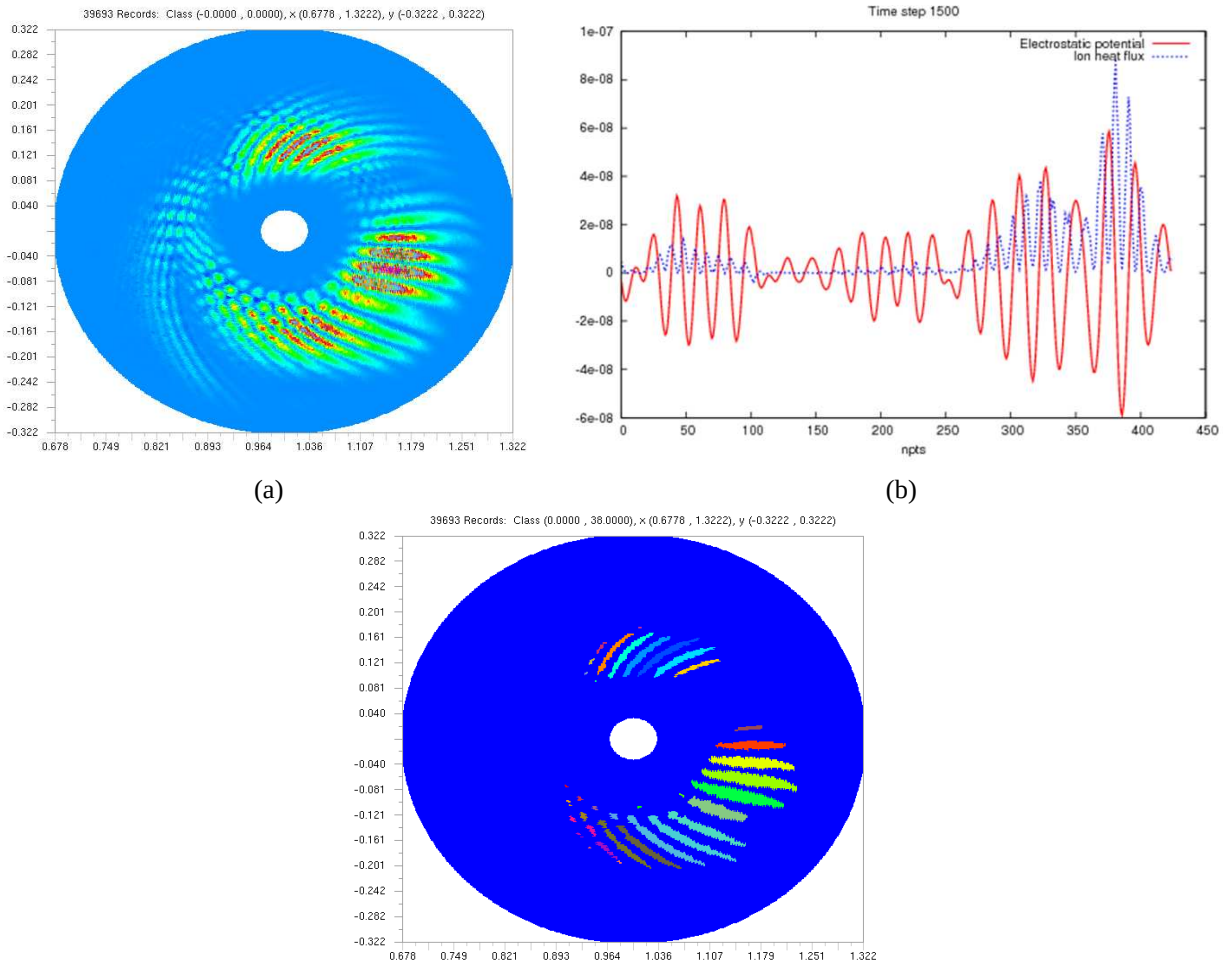

FIG. 8: Identifying the coherent structures in the ion heat flux variable in a two-dimensional slice of a fusion simulation. (a) The variable values at time step 1500. Grid points with similar colors have similar values, but the colors themselves do not have any significance. (b) The plot of the electrostatic potential (in red, continuous line) and the ion heat flux (in blue, dotted line) along a flux surface [the grid points that lie on a circle in the panel (a)]. (c) The structures identified in the data, with all points in a structure assigned the same color.

5. EXTRACTING FEATURES TO REPRESENT THE OBJECTS

Once we have identified and extracted the objects of interest in the data we need to represent them using features, which are low level measurements extracted from the data. This step is obviously very problem dependent. For example, if we are interested in identifying galaxies with a bent-double morphology, we would extract features such as angles between the blobs that represent a galaxy (see Fig. 11, left panel). On the other hand, if we were interested in galaxies with a different shape, we would extract other features representative of that shape.

Identifying appropriate features that are representative of the patterns we seek in the data is not the only challenge in feature extraction. We also need to identify features that are scale, rotation, and translation invariant as the patterns themselves are often scale, rotation, and translation invariant. For example, in our work on comparing simulations and experiments of Richtmyer-Meshkov instability, we compared the two by extracting statistics on the mushroom-shaped

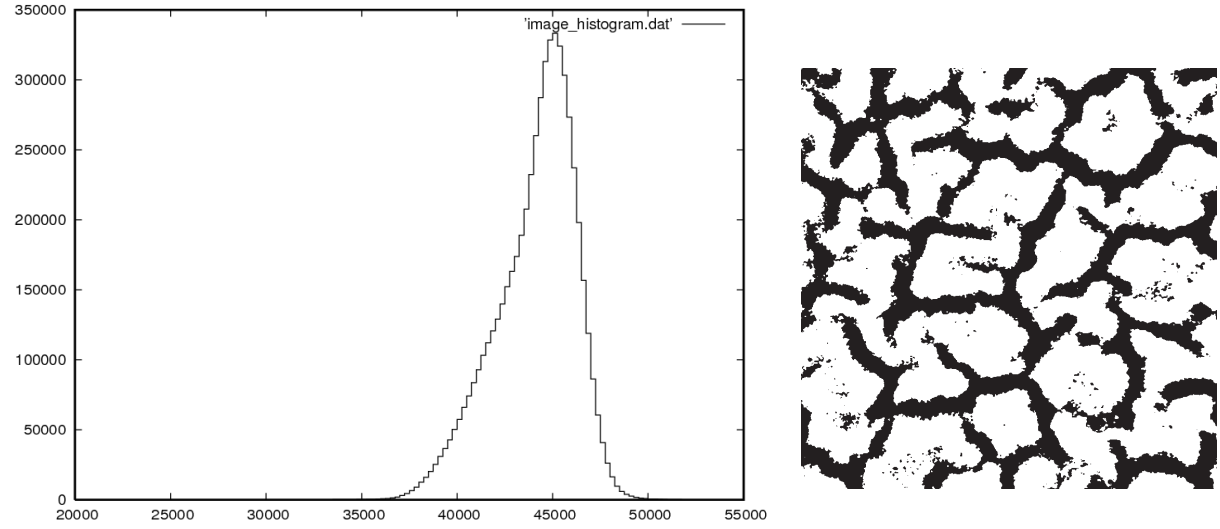

FIG. 9: Segmenting the image from the fragmentation of materials problem. Left: the histogram of the image; note the asymmetry. Right: the image obtained after thresholding, with the threshold selected at the point of asymmetry. The black pixels represent the gap region from the original image. Note the spurious pixels that must be removed by post-processing.

structures in the data. As shown in Fig. 10, these statistics include the height of the mushroom, the width and height of the mushroom cap, the width of the mushroom stem, and so on. However, as the resolutions of the simulation and experimental data were different, we could not use these features directly for comparison. An alternative was to use ratios of the primary features, which would now be independent of the scale.

Another important issue in feature extraction is that we must extract the features in a robust way; that is, the feature values must not be sensitive to small changes in the data. For example, we initially considered using the position angle as a feature in our work on identifying bent-double galaxies in the FIRST astronomical survey. However, one of the astronomers pointed out that this feature was not robust (as shown in Fig. 11, right panel) since the position angle, which is measured counter-clockwise from North, could change by a large amount with a slight change in the direction of the elliptic Gaussian.

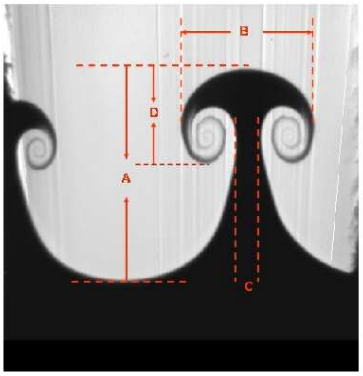

FIG. 10: Features representing the mushroom-shaped structures in the problem of code validation Richtmyer-Meshkov simulations. These features include the height of the mushroom, the width and height of the mushroom cap, and the width of the mushroom stem.

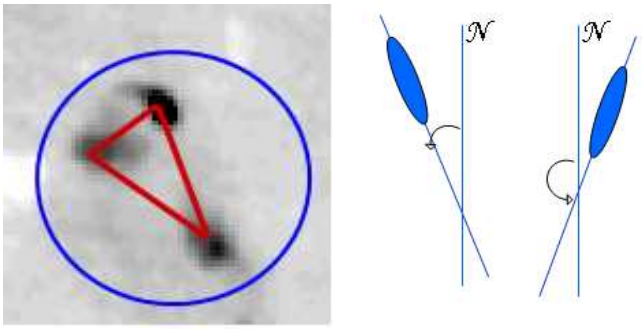

FIG. 11: Extracting features for the problem of classification of bent-double galaxies in the FIRST astronomical survey: left: the grouping of the blobs in the catalog (the blue circle) to identify a galaxy and extract distances and angle features (the red triangle); right: the lack of robustness of the position angle as a feature—a small change in the direction of an elliptic Gaussian can lead to a large change in the position angle.

6. REDUCING THE DIMENSION OF THE PROBLEM

Once the features for the objects in the data have been extracted, we can write the data set as an object by feature matrix, where each row in the matrix corresponds to an object described by its features in the columns. The number of features is the dimension of the problem since each object can be represented as a point in a multi-dimensional feature space. Often, each object is described using a large number of features, which can number in the tens or hundreds. This is especially true if the data are multi-variate; for example, multi-spectral data in remote sensing.

From a data mining viewpoint, there are several reasons why we may want to consider reducing the dimension of a problem [1]. In some problems, we may want to identify the important variables relevant to a phenomenon since we would like to monitor these variables to understand the phenomenon better. In the case of classification problems, adding irrelevant features to the data can reduce the accuracy of decision tree classifiers, which could focus on these irrelevant features if there are a few samples at a node of the tree. Also, from the viewpoint of the "curse of dimensionality," more samples are required to cover a high-dimensional space adequately, leading to a need for larger training sets in classification problems and a loss in the meaning of the term "nearest neighbors" in clustering problems. We also may need to consider the additional cost to extract, store, and use the irrelevant features, especially since data structures for fast searches do not scale to high dimensions. The latter is especially important in problems involving information retrieval or nearest-neighbor searches. And, finally, lower-dimensional data sets are easier to visualize and understand.

There are two broad categories of dimension reduction algorithms commonly used in data mining—the feature transform methods and the feature selection methods. The former transform the current set of features into another set in a lower-dimensional space. The latter category of techniques select a subset of the current features and often are used when we need to maintain the connection to the original features. We next discuss these methods briefly.

6.1 Feature Transform Methods

One of the most commonly used feature transform methods is principal component analysis (PCA) [33], whose variants are referred to by many names including Karhunen-Loève transform, empirical orthogonal functions, Hotelling transform, proper orthogonal decomposition, latent semantic indexing, and singular value decomposition. The goal in PCA is to use linear combinations of the input features to transform them into a set that is uncorrelated and where only a few features are necessary to adequately describe the data. These derived features are called principal components. In essence, PCA is the orthogonal projection of the data onto a lower-dimensional space such that the variance of the projected data is maximized.

Among all the linear techniques, PCA gives the minimum mean-squared-error approximation in the reduced dimension space. This is good for tasks such as the compression of the data, but it does not necessarily lead to maximum

class separability in the lower-dimension space, which can make it unsuitable for classification tasks. Furthermore, PCA finds a linear subspace and, therefore, cannot handle data lying on nonlinear manifolds.

Several techniques have been proposed to address these deficiencies [34]. Many of these are non-linear generalizations of PCA, such as projection pursuit [35, 36], principal curves and surfaces [37, 38], and kernel PCA [39]. More recently, there has been active research in techniques for the case where the data lie on a nonlinear manifold in the lower-dimensional space [40]. These non-linear manifold learning techniques include methods such as Isomap, locally linear embedding (LLE), and Laplacian eigenmaps, to name just a few. The basic idea is to identify a transformation that preserves some quantity, such as the geodetic distance between two points in Isomap, the local neighborhood relations in LLE, or the pairwise distance in Laplacian eigenmaps,

An interesting transform-based technique for dimension reduction is random projections [41], where the original high-dimensional data are projected onto a lower-dimensional subspace using a random matrix whose columns have unit length. Underlying random projections is the Johnson-Lindenstrauss lemma, which states that any set of points of dimension n in a Euclidean space can be embedded in $O(\log n/\epsilon^2)$ dimensions without distorting the distances between any pair of points by more than a factor of $(1 \pm \epsilon)$ for any $0 < \epsilon < 1$. As the method is computationally very efficient, it can be used prior to the application of the more compute intensive feature transform methods.

6.2 Feature Selection Methods

The idea of selecting a subset of the features arose in the machine learning community as a means of reducing the dimension of the problem. These methods are applicable in classification and regression problems, where there is a discrete or continuous output variable associated with each object, and the intent is to build a predictive model to predict this variable. There are two categories of feature selection techniques—filters and wrappers.

Filter methods select important features based on how well they discriminate among the different classes. For example, the Kullback-Leibler (KL) class separability filter [42] calculates the class separability of each feature using the KL distance between histograms of feature values. For each feature, there is one histogram for each class. We discretize the numeric features using $\sqrt{|D|}/2$ equally spaced bins, where $|D|$ is the size of the data. The histograms are normalized by dividing each bin count by the total number of elements to estimate the probability, $p_j(d = i|c = n)$, that the jth feature takes a value in the ith bin of the histogram given a class n. For each feature j, we can then calculate the class separability as

$$\Delta_j = \sum_{m=1}^{c} \sum_{n=1}^{c} \delta_j(m, n),$$

where c is the number of classes and $\delta_j(m, n)$, the KL distance between histograms corresponding to classes m and n, is given by

$$\delta_j(m, n) = \sum_{i=1}^{b} p_j(d = i|c = m) \log \left(\frac{p_j(d = i|c = m)}{p_j(d = i|c = n)} \right),$$

where b is the number of bins in the histograms. If the histograms have little overlap—that is, the distance is large—then the feature is a good feature and can be used to discriminate between the classes.

The chi-squared filter computes the chi-square statistics from contingency tables for every feature. These tables have a row for every class label and the columns correspond to possible values of the feature (see Table 1, data adapted from [43]). Numeric features are represented by histograms, so the columns of the contingency table are the histogram bins. The chi-square statistic for feature j is

$$\chi_j^2 = \sum_i \frac{(o_i - e_i)^2}{e_i},$$

where the sum is over all the cells in the $r \times c$ contingency table; r is the number of rows; c is the number of columns; o_i stands for the observed value (the count of the items corresponding to the cell i in the contingency table); and e_i is the expected frequency of items calculated as

TABLE 1: A 2×3 contingency table, with observed and expected frequencies (in parentheses), of a fictitious feature fl that takes three possible values (1, 2, and 3).

Class	f1=1	f1=2	f1=3	Total
0	31 (22.5)	20 (21)	11 (18.5)	62
1	14 (22.5)	22 (21)	26 (18.5)	62
Total	45	42	37	124

$$e_i = \frac{(\text{column total}) \times (\text{row total})}{\text{grand total}}$$

The variables are ranked by sorting them in descending order of their χ^2 statistics.

Some classification techniques also provide an indication of which features are important. For example, a stump filter [42] is derived from a decision tree [44] by using the same process as the one used to create the root node of the tree (hence, the name "stump"). Decision trees split the data by examining each feature and finding the split that optimizes an impurity measure. To search for the optimal split for a numeric feature x, the feature values are sorted ($x_1 < x_2 < ... < x_n$) and all intermediate values $(x_i + x_{i+1})/2$ are evaluated as possible splits using a given impurity measure. The features are then ranked according to their optimal impurity measures. We use the Gini criterion, which is based on finding the split that most reduces the node impurity, where the impurity is defined as follows:

$$L_{\text{Gini}} = 1.0 - \sum_{i=1}^{k} (L_i/|T_L|)^2, \quad R_{\text{Gini}} = 1.0 - \sum_{i=1}^{k} (R_i/|T_R|)^2$$

$$\text{Impurity} = (|T_L| * L_{\text{Gini}} + |T_R| * R_{\text{Gini}})/n$$

where $|T_L|$ and $|T_R|$ are the number of examples, and L_{Gini} and R_{Gini} are the Gini indices on the left and right sides of the split, respectively.

In contrast to the filter methods, which are independent of the classification or regression method used subsequent to the feature selection process, the wrapper methods incorporate the technique used to build the predictive model. They consider candidate feature subsets, evaluate them using the classification or regression algorithm, and select the subset that yields the most accurate results. For computational efficiency, the candidate feature subsets are selected using a greedy algorithm. We can either start by selecting the best single feature, finding the next best feature to add, and continuing the process of growing the subset until the accuracy of the model built using the selected features no longer increases. Or, we can start by selecting all the features, and removing features that do not contribute to the accuracy of the model. Since the predictive models have to be built and evaluated several times, wrapper methods are more computationally expensive than the filter methods.

6.3 Observations on Dimension Reduction Techniques

Our experiences with dimension reduction techniques, in the context of several practical problems, have indicated that feature selection techniques tend to work better than feature transform methods. Figure 12 illustrates some of our results on two classification problems. In the plots, we show how the error rate for classification, obtained using five fold cross validation repeated five times, varies when we use the top k features selected by different dimension reduction techniques to build the classifier. The classifier used is a decision-tree-based approach [45], which creates ensembles by introducing randomization at each node of the tree in two ways. It first randomly samples the examples at a node and selects a fraction (we use 0.7) for further consideration. Then, for each feature, instead of sorting these examples based on the values of the feature, it creates a histogram, evaluates the splitting criterion (we use Gini [44]) at the mid-point of each bin of the histogram, identifies the best bin, and then selects the split point randomly in this bin. The randomization is introduced both in the sampling and in the choice of the split point. We use 10 trees in the ensemble. The horizontal line in each plot is the error rate obtained using all the features.

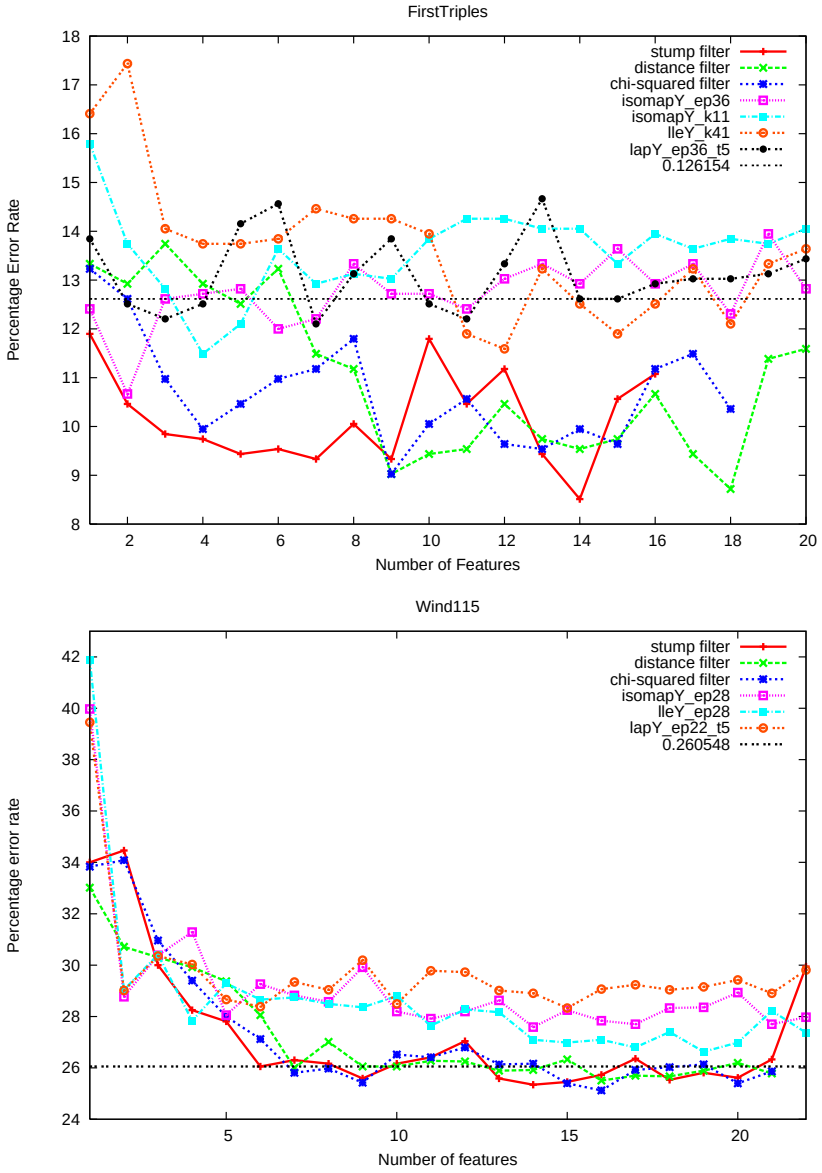

FIG. 12: The variation in the percentage error rate for classification, obtained using fivefold cross-validation repeated five times, as a function of the top k features selected by different dimension reduction techniques. Top: the classification of bent-double galaxies in the FIRST astronomical survey. Bottom: the classification of days with ramp events in a wind power generation problem.

The top panel in Fig. 12 shows the results for the classification of bent-double galaxies in the FIRST survey for the case of galaxies composed of three blobs or catalog entries. This data set is quite small, consisting of 195 examples, with 167 bents and 28 non-bents. Each galaxy is described by 20 numeric features obtained by considering the three Gaussians representing the three blobs.

The bottom panel in Fig. 12 shows the results for a problem involving the prediction of ramp events in wind power generation [46]. A ramp event occurs when the wind power generation suddenly increases or decreases by a large amount in a short time. These events make it difficult for the control room operators to schedule wind energy on the

power grid. Our analysis task in this problem was to determine if we could use weather conditions to predict if a day is likely to have a ramp event. In this data set, we have 731 examples representing the data for the days in 2007–2008. The features are the daily averages of different variables—such as wind speed, wind direction, and temperature—at three meteorological towers in the region of the wind farm. Each tower provides seven features, for a total of 21 features. Assigned to each day is a binary class variable that indicates if a ramp event exceeding 115 MW occurred in any 1 hour interval during that day.

These plots show that the feature selection techniques (KL distance filter, stump filter, and chi-squared filter) tend to give lower error rates than the non-linear manifold learning techniques (isomap, LLE, and Laplacian eigenmaps). In both problems, we see that the non-linear transform methods can often give worse error rates than using all the original features (indicated by the horizontal line). This could indicate that the data for our problems do not lie on a non-linear manifold. The feature selection methods, especially the filter methods, are also computationally faster than the transform methods. In addition, as they select a subset of the features, the results are interpretable, which is important as it can provide scientific insights into the data.

In Fig. 13, we show the results of the application of random projections to a data set obtained from a simulation of the chlorine concentration in a water distribution network [47]. The data for 166 sensors are available for 15 days of simulation. They are collected once every 5 min, for a total of 4310 time instants. In Fig. 13, we show the results using a subset of the data representing 5 days (1440 instants). We use one of the random matrices, R, defined in [41], whose elements r_{ij} are chosen from the following simple probability distribution:

$$r_{ij} = \sqrt{3} \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases}$$

The random projection is then obtained by multiplying the input matrix A, of order $n \times d$, which represents the n d-dimensional data points, by R. Here, $d = 166$, the original dimension of the chlorine data. The plot in Fig. 13 is obtained by considering different values of k and evaluating the distortion resulting from the projection. The distortion is defined as:

$$\frac{1}{k} \frac{\|\mu(x) - \mu(y)\|_2}{\|x - y\|_2}$$

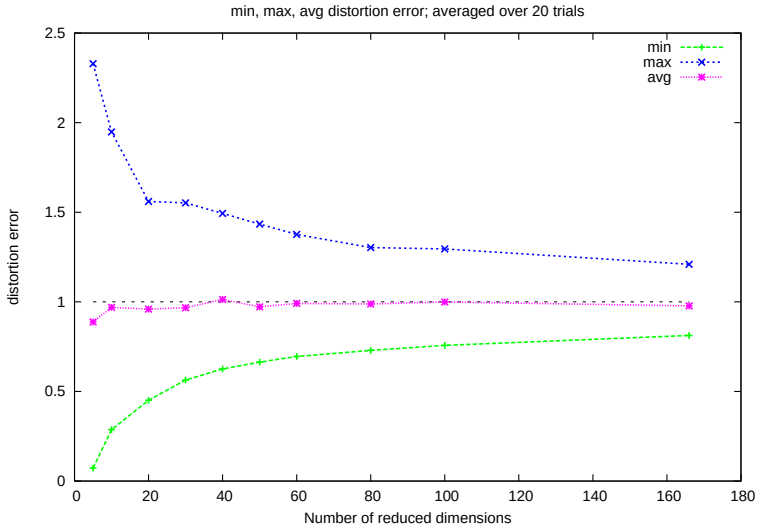

FIG. 13: The distortion using random projection as the number of reduced dimensions is varied; the maximum, minimum, and average distortions are the average of these quantities over 20 tries.

where $\mu(x)$ is the projection of the vector x. Figure 13 shows the results of the maximum, average, and minimum distortion as the value of k is increased from 10 to 166. The result for each value of k is the average over 20 random matrices.

This plot shows that the average distortion, even when the number of reduced dimensions is small, is close to 1.0, indicating that on an average, the distances are preserved when the points are projected onto the lower-dimensional space. However, the maximum and minimum distortions, averaged over 20 tries, can be quite far from 1.0 when the number of reduced dimensions is low, but reduce rapidly as this number increases. Also, the maximum and minimum distortion in an 80-dimensional space is close to the maximum and minimum distortion using all the 166 variables. Since the random projections can be easily obtained, they provide a simple and cost-effective way of reducing the number of dimensions for problems where the reduced dimensional data are used in distance-based algorithms, such as nearest-neighbor methods. The "instability" of random projections also can be exploited by considering several random matrices for the projection and using ensemble approaches in the analysis.

7. BUILDING MODELS FROM THE DATA

The focus in data mining literature often tends to be on the process of building models, with many new algorithms and untold number of variations on old ones being proposed for tasks such as classification, clustering, anomaly detection, and association rules. In our experience, it is the pre-processing of the data that is done prior to the building of models that is more critical to the success of a data mining endeavor. So, instead of focusing on any specific algorithms, we next briefly describe the types of models that are built in scientific problems.

7.1 Scaling Laws

In some problems, where high-performance computers are used to run first principle calculations, scientists are often interested in identifying scaling laws in their data. For example, in our problem of the analysis of large eddy and direct numerical simulations of the Rayleigh-Taylor instability, scientists were interested in determining how the count of the bubble and spike structures varied over time. Figure 14 shows the count of the number of bubbles with time using different methods for the direct numerical simulation of the Rayleigh-Taylor instability [28]. The similarity of the plots

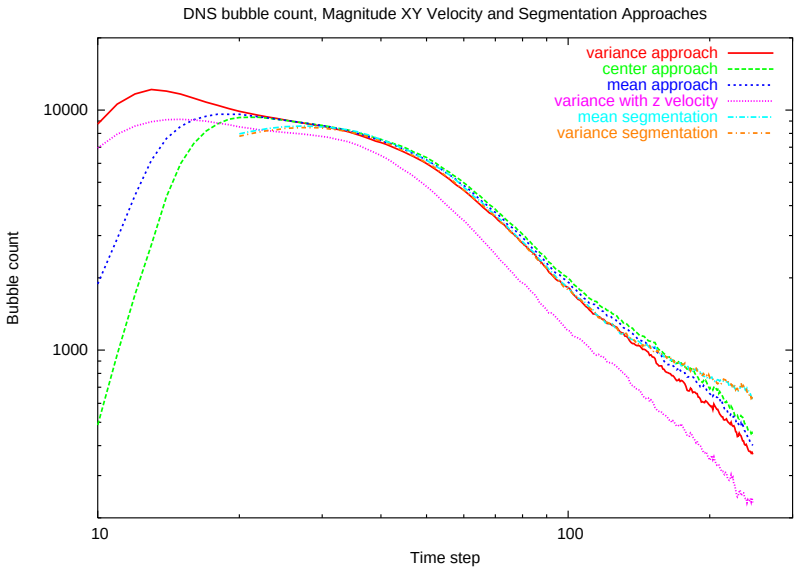

FIG. 14: The count of the number of bubbles obtained using different methods for the direct numerical simulation of the Rayleigh-Taylor instability.

from different methods gives greater confidence in the results. The plots also show that there are four distinct regimes in the process of fluid mixing—an initial linear growth where the initial perturbation at the fluid interface grows in magnitude, with the bubbles growing independent of each other; a second non-linear stage where some bubbles grow faster than the others; a third phase of mixing transition; and the final phase of strong turbulence, leading to well-mixed fluid. The slopes of the curves in each of these phases are important as they can be used to provide a succinct power-law representation of the count. However, fitting power-law distributions to empirical data must be done with care to ensure that the distribution is indeed a power-law distribution and to calculate the scale factors accurately [48].

7.2 Predictive Models

Here, the goal is to build a model from a training set consisting of objects, their features, and an output, which could be discrete or continuous, and then use the model to predict an output for objects described by their features. If the output is discrete, the problem is one of classification, if it is continuous, the problem is a regression problem. There are a host of methods for building both kinds of models, ranging from classification and regression trees, support vector machines, neural networks, locally weighted learning, and so on [49–52]. An important development in the last decade in building more accurate predictive models has been the work in ensemble learning. Here, a trade-off is made between the bias and variance of the models by incorporating randomization to create a series of models (that is, the ensemble) from the same training set and then obtaining a prediction by suitably combining the predictions of these models [53].

There are some factors that must be considered in the creation and application of predictive models. In some problems, interpretable models, such as those generated by decision and regression trees, may be important for the insights they provide in how they arrive at the output variable. We also have found that it is worthwhile to spend the effort to create a high-quality training set. A large number of incorrect labels can give inaccurate results, while an unbalanced training set, with a majority of examples from one class, can give rise to artificially high accuracy in cross-validation results.

7.3 Descriptive Models

These models are built using clustering techniques where we use the features representing the objects to group similar ones together. There are numerous clustering approaches, including the iterative k-means, bottom up agglomerative methods, top-down divisive methods, graph-based approaches, and spectral methods [52, 54–58]. The results of clustering are often dependent on the number of clusters requested and the definition of the similarity metric, which identifies the similarity between two objects.

8. GENERATING THE DATA TO BE ANALYZED

Until the last decade or so, when we consider the different applications of data mining techniques, we find that the analysis of the data was usually done after the data were collected. In contrast, in some fields, such as statistics, there always has been a well-developed approach to the "design of experiments," where much thought was given to the collection of the data themselves [59].

As data mining techniques are being applied to a diverse set of problems, it is becoming clear that we also need to pay attention to the process of generating the data so we can close the gap between data acquisition and model building. There are several reasons why this is desirable. First, it may help improve the quality of the data. For example, having data miners be involved in the generation of a training set may result in a well-balanced set, instead of one where objects of the class of interest to the domain scientists far out number the other classes, which may be of little interest. If new examples have to be assigned labels, and this process is time consuming or tedious, we could use data mining techniques, such as active learning [60, 61], to judiciously identify the new examples to be labeled so that we can learn a more accurate model using few additional examples. Other techniques that are helpful in building better training sets include semi-supervised learning [62] or the incorporation of relevance feedback ideas from information retrieval to evaluate the results from an initial application of the classifier.

The second broad area where data mining techniques can help in the generation of the data is in the area of design of computer experiments [63]. As scientists run a series or an ensemble of computer simulations for sensitivity analysis, or uncertainty quantification, to simulate how an experiment would perform under certain conditions, to understand the design space for the creation of new materials, they are increasingly faced with the problem of what values to use for the input parameters to their simulations. Since the simulations are expensive, they would like to run just a few of them. At the same time, they would like these simulations to meet certain criteria. For example, if the goal is to explore a space of possible designs, we need to cover the space well so we can identify global optima. On the other hand, if we are interested in seeing how an experiment will perform with certain inputs, the problem is more localized, and we should sample the input space of the simulations so that there is enough variation in the ensemble to account for uncertainties in the simulation and the experimental inputs. In both these cases, sampling techniques from statistics and design of experiments are relevant. Also, we can use feature selection techniques to determine the important inputs for use in generating the samples. In addition, code surrogates or meta-models ([1], pp. 29), which are essentially models built using classification or regression methods, can be used to predict the output for a given set of input parameters. Of course, the training set used to build these models must be of high quality, both in coverage and in accuracy, for the meta-model to be truly predictive. We also observe that some of these problems can be phrased as inverse problems where the task is to find the inputs to a simulation that lead to a desired output; here, ideas from landscape characterization can be helpful [64].

9. SUMMARY AND CONCLUDING THOUGHTS

In this paper, we described the data mining process and, using example problems, described how the tasks in the process are being used in the analysis of scientific data sets in a variety of domains. These techniques can be relevant in several contexts in stochastic modeling and uncertainty quantification. These range from automating the analysis of simulation output so that ensembles of simulations can be analyzed easily; the use of dimension reduction to identify important variables and to map the data to a lower-dimensional space; the building of data-driven models, both predictive and descriptive; and the generation of the data themselves. Although not explicitly mentioned in the paper, techniques for making decisions under uncertainty have long been studied in fields that preceded data mining under topics such as “reasoning under uncertainty” and “uncertainty in artificial intelligence”; these topics are seeing a resurgence as scientists in various application domains exploit the abundance of data and utilize them, both for insights and in decision support.

While data mining techniques can contribute to stochastic modeling and uncertainty quantification, we would be remiss if we did not mention the opportunity for a two-way exchange of ideas. In particular, the approaches for sampling a parameter space using multi-level Monte Carlo or quasi-Monte Carlo methods, the use of Gaussian process regression and sparse grids, as well as the approaches taken to address stochasticity and uncertainty in the solution of practical problems in science and engineering, are all worthy of further investigation for ways in which they can enhance traditional data mining.

Finally, we end with a few notes of caution regarding the application of data mining techniques and mention some of the current areas of active research. Scientific data mining is a careful and considered application of techniques in close collaboration with domain scientists. It is an iterative and interactive process. The pre-processing of the data, although time consuming, is crucial to the success of the process. While it may be tempting to do so, we do not recommend that the techniques for the different tasks in the data mining process be blindly applied to the data. It often helps to try different techniques to see if they give the same results. Many different fields contribute to data mining, each providing its own perspectives and insights into a problem and a solution approach. The field of data mining in general, and scientific data mining in particular, are actively growing to accommodate the needs of new problems and data types. Real-time analysis of multi-variate sensor data streams, especially in the presence of concept drift, is becoming increasingly important. The size of the data also is becoming larger and more complex, especially for experiments and observations, where multi-variate, multi-sensor, multi-modal data are becoming the norm. In the case of simulations, the move toward exascale systems, with their limited I/O bandwidth, is causing concern since it is unclear if we can move all the analysis tasks in situ to avoid writing out large volumes of data. There are new analysis problems that arise as we consider bridging the gap between data acquisition and analysis. As data mining techniques

mature and are applied successfully to real problems, they are increasingly being considered as part of a loop in the process of decision support, resulting in reasoning under uncertainty and uncertainty quantification becoming an integral part of the data mining process.

ACKNOWLEDGMENTS

There are several individuals who contributed to the work presented in this paper. I would like to thank the data miners and the software developers—Erick Cantú-Paz, Samson Sen-Ching Cheung, Ya Ju Fan, Abel Gezahegne, Thinh Nguyen, and Nu Ai Tang—who collaborated with me in the development of the software and the analysis of the data. Our work would not have been possible without the interest of the domain scientists who shared their data and their expertise. I gratefully acknowledge Robert Becker and the FIRST astronomers, Jeffrey Greenough, Omar Hurricane, Jeffrey Jacobs, Zhihong Lin, and Paul Miller, for being so gracious with their time. The work presented in this paper was supported over the years by the ASC program at the Department of Energy (DOE), the ASCR Basic Research Program at DOE, the LDRD program at Lawrence Livermore National Laboratory, the SciDAC program at DOE, and the WHTP program at EERE at DOE—their financial support of this work is gratefully acknowledged. More details on the problems and analysis presented in this paper can be found at the following two web sites: https://computation.llnl.gov/casc/StarSapphire/ and https://computation.llnl.gov/casc/sapphire/. This work (LLNL-JRNL-496765) has been performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344.

REFERENCES

1. Kamath, C., *Scientific Data Mining: A Practical Perspective*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.
2. Sivia, D. S. and Skilling, J., *Data Analysis: A Bayesian Tutorial*, Oxford University Press, Oxford, UK, 2006.
3. Jordan, M., ed., *Learning in Graphical Models*, The MIT Press, Cambridge, MA, 1998.
4. Koller, D. and Firedman, N., *Probabilistic Graphical Models: Principles and Techniques*, The MIT Press, Cambridge, MA, 2009.
5. Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.
6. Sonka, M., Hlavac, V., and Boyle, R., *Image Processing, Analysis, and Machine Vision*, International Thompson Publishing, Pacific Grove, CA, 1999.
7. Weeratunga, S. K. and Kamath, C., PDE-based non-linear diffusion techniques for denoising scientific and industrial images: An empirical study, *Proc. of Image Processing: Algorithms and Systems, SPIE Electronic Imaging Symposium*, vol. 4667, pp. 279–290, SPIE Press, Bellingham, WA, 2002.
8. Weeratunga, S. K. and Kamath, C., A comparison of PDE-based non-linear anisotropic diffusion techniques for image denoising, *Proc. of Image Processing: Algorithms and Systems, SPIE Electronic Imaging Symposium*, vol. 5014, pp. 201–212, SPIE Press, Bellingham, WA, 2003.
9. Winkler, G., *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*, Springer, Berlin, 2006.
10. Fieguth, P., *Statistical Image Processing and Multidimensional Modeling*, Springer, Berlin, 2010.
11. Brouillette, N., The Richtmyer-Meshkov instability, *Ann. Rev. Fluid Mech.*, 34:445–468, 2002.
12. Jacobs, J. W. and Collins, B. D., Experimental study of the Richtmyer-Meshkov instability of a diffuse interface, *Proc. of 22nd International Symposium on Shock Waves*, July, 1999.
13. Kamath, C. and Miller, P. L., Image analysis for validation of simulations of a fluid-mix problem, *Proc. of IEEE International Conference on Image Processing*, vol. III, pp. 525–528, 2007.
14. Kokaram, A. C., *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, Springer, Berlin, 1998.
15. Kokaram, A. C., Removal of line artefacts for digital dissemination of archived film and video, *Proc. of the IEEE Conference on Multimedia Computing and Systems*, vol. 2, pp. 245–249, 1999.

16. Land, E., An alternative technique for the computation of the designator in the retinex theory of color vision, *Proc. Natl. Acad. Sci.*, 83:3078–3080, 1986.
17. Jobson, D. J., Rahman, Z., and Woodell, G. A., A multi-scale retinex for bridging the gap between color images and the human observation of scenes, *IEEE Trans. Image Proc.*, 6(7):965–976, 1997.
18. Rahman, Z., Jobson, D. J., and Woodell, G. A., Retinex processing for automatic image enhancement, *J. Electron. Imaging*, 13(1):100–110, 2004.
19. Kamath, C. and Hurricane, O. A., Robust extraction of statistics from images of material fragmentation, *Int. J. Image Graph.*, 11:377–401, 2011.
20. Canny, J., A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.
21. Smith, S. M. and Brady, J., SUSAN—A new approach to low-level image processing, *Int. J. Comput. Vis.*, 23(1):45–78, 1997.
22. Weeratunga, S. K. and Kamath, C., Investigation of implicit active contours for scientific image segmentation, *Proc. of Visual Communications and Image Processing*, vol. 5308, pp. 210–221, SPIE Press, Bellingham, WA, 2004.
23. Cook, A. W., Cabot, W. H., and Miller, P. L., The mixing transition in Rayleigh-Taylor instability, *J. Fluid Mech.*, 511:333–362, 2004.
24. Cabot, W. H. and Cook, A. W., Reynolds number effects on Rayleigh-Taylor instability with possible implications for type Ia supernovae, *Nat. Phys.*, 2:562–568, 2006.
25. Kamath, C., Gezahegne, A., and Miller, P. L., Analysis of Rayleigh-Taylor instability, Part I: Bubble and spike count, Tech. Rep. UCRL-TR-223676, Lawrence Livermore National Laboratory, 2006.
26. Kamath, C., Gezahegne, A., and Miller, P. L., Analysis of Rayleigh-Taylor instability: Statistics on rising bubbles and falling spikes, Tech. Rep. UCRL-TR-236111-REV-1, Lawrence Livermore National Laboratory, 2007.
27. Gezahegne, A. and Kamath, C., Tracking non-rigid structures in computer simulations, *Proc. of IEEE International Conference on Image Processing*, pp. 1548–1551, 2008.
28. Kamath, C., Gezahegne, A., and Miller, P. L., Identification of coherent structures in three-dimensional simulations of a fluid-mix problem, *Int. J. Image Graph.*, 9:389–410, 2009.
29. Becker, R. H., Helfand, D. J., White, R. L., Gregg, M. D., and Laurent-Muehleisen, S. A., Faint Images of the Radio Sky at Twenty Centimeters (FIRST), URL: http://sundog.stsci.edu, 2011.
30. Kamath, C., Cantú-Paz, E., Fodor, I. K., and Tang, N. Searching for bent-double galaxies in the first survey, *Data Mining for Scientific and Engineering Applications*, Grossman, R., Kamath, C., Kegelmeyer, W. P., Kumar, V., and Namburu, R. (eds.), pp. 95–114, Kluwer, Boston, MA, 2001.
31. Lin, Z., Chen, L., Heidbrink, W. W., Waltz, R. E., Spong, D., and Kamath, C., Gyrokinetic Simulation of Energetic Particle Turbulence and Transport (GSEP) Project, URL: http://phoenix.ps.uci.edu/gsep/, 2011.
32. Otsu, N., A threshold selection method from gray-level histograms, *IEEE Trans. Syst., Man Cybern.*, 9(1):62–66, 1979.
33. Jolliffe, I. T., *Principal Components Analysis*, 2nd ed., Springer, New York, 2002.
34. Carriera-Perpiñán, M. A., Continuous latent variable models for dimensionality reduction and sequential data reconstruction, PhD thesis, University of Sheffield, UK, 2001 (see chapter 4 on dimension reduction).
35. Friedman, J. H. and Tukey, J. W., A projection pursuit algorithm for exploratory data analysis, *IEEE Trans. Comput.*, C-23(9):881–890, 1974.
36. Huber, P. J., Projection pursuit, *Ann. Stat.*, 13(2):435–475, 1985.
37. Hastie, T., Principal curves and surfaces, *Technical Report SLAC-0276*, Stanford Linear Accelerator Center, 1984.
38. Hastie, T. and Stuetzle, W., Principal curves, *J. Am. Stat. Assoc.*, 84:502–516, 1989.
39. Schölkopf, B., Smola, A., and Müller, K.-R., Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.*, 10(5):1299–1319, 1998.
40. van der Maaten, L. J. P., Postma, E. O., and van den Herik, H. J., Dimensionality reduction: A comparative review, *Technical Report TiCC TR 2009–005*, Tilburg University, 2009.
41. Achlioptas, D., Database-friendly random projections: Johnson-Lindenstrauss with binary coins, *J. Comput. System Sci.*, 66:671–687, 2003.
42. Cantú-Paz, E., Newsam, S., and Kamath, C., Feature selection for scientific applications, *Proc. of the SIGKDD International*

Conference on Knowledge Discovery and Data Mining, pp. 788–793, 2004.

43. Huang, S. H., Dimensionality reduction on automatic knowledge acquisition: A simple greedy search approach, *IEEE Trans. Knowl. Data Eng.*, 15(6):1364–1373, 2003.
44. Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., *Classification and Regression Trees*, CRC Press, Boca Raton, FL, 1984.
45. Kamath, C., Cantú-Paz, E., and Littau, D., Approximate splitting for ensembles of trees using histograms, *Proc. of 2nd SIAM International Conference on Data Mining*, pp. 370–383, 2002.
46. Kamath, C., Associating weather conditions with ramp events in wind power generation, *Proc. of IEEE PES Power Systems Conference and Exposition*, 2011.
47. Kamath, C., Dimension reduction for streaming data, *Data Intensive Computing: Architectures, Algorithms, and Applications*, Gorton, A. and Gracio, D., eds., Cambridge University Press, Cambridge, UK, 2012.
48. Clauset, A., Shalizi, C. R., and Newman, M. E. J., Power-law distributions in empirical data, *SIAM Rev.*, 51(4):661–703, 2009.
49. Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, Wiley, New York, 2001.
50. Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2001.
51. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
52. Theodoridis, S. and Koutroumbas, K., *Pattern Recognition*, Academic Press, San Diego, CA, 2006.
53. Rokach, L., Ensemble-based classifiers, *Artif. Intell. Rev.*, 33(1-2):1–39, 2010.
54. Karypis, G. and Kumar, V., Multilevel k-way partitioning scheme for irregular graphs, *J. Parallel Distrib. Comput.*, 48(1):96–129, 1998.
55. Karypis, G. and Kumar, V., A fast and high-quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.*, 20(1):359–392, 1999.
56. Xu, R. and Wunsch, D., Survey of clustering algorithms, *IEEE Trans. Neural Netw.*, 16(3):645–678, 2005.
57. Gan, G., Ma, C., and Wu, J., *Data Clustering: Theory, Algorithms, and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007.
58. Luxborg, U., A tutorial on spectral clustering, *Stat. Comput.*, 17(4):395–416, 2007.
59. Montgomery, D. C., *Design and Analysis of Experiments*, Wiley, Hoboken, NJ, 2004.
60. Iyengar, V. S., Apte, C., and Zhang, T., Active learning using adaptive resampling, *Proc. of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 92–98, 2000.
61. Settles, B., Active learning literature survey, *Technical Report 1648*, Computer Science Department, University of Wisconsin-Madison, 2010.
62. Zhu, X., Semi-supervised learning literature survey, *Technical Report 1530*, Computer Science Department, University of Wisconsin-Madison, 2008.
63. Fang, K.-T., Li, R., and Sudjianto, A., *Design and Modeling for Computer Experiments*, Chapman & Hall/CRC Press, Boca Raton, FL, 2005.
64. Burl, M. C., DeCoste, D., Enke, B. L., Mazzoni, D., Merline, W. J., and Scharenbroich, L., Automated knowledge discovery from simulators, *Proc. of 6th SIAM International Conference on Data Mining*, pp. 82–93, 2006.