SAMPLING STRATEGIES FOR MINING IN DATA-SCARCE DOMAINS

A novel framework leverages physical properties for mining in data-scarce domains. It interleaves bottom-up data mining with top-down data collection, leading to effective and explainable sampling strategies.

Several important scientific and engineering applications require analysis of spatially distributed data from expensive experiments or complex simulations, which can demand days, weeks, or even years on petaflops-class computing systems. Consider the conceptual design of a high-speed civil transport, which involves the disciplines of aerodynamics, structures, mission-related controls, and propulsion (see Figure 1).[1] Frequently, the engineer will change some aspect of a nominal design point and run a simulation to see how the change affects the objective function (for example, take-off gross weight, or TOGW). Or the design process is made configurable, so the engineer can concentrate on accurately modeling one aspect while replacing the remainder of the design with fixed boundary conditions surrounding the focal area. However, both these approaches are inadequate for exploring large high-dimensional design spaces, even at low fidelity. Ideally, the design engineer would like a high-level mining system to identify the pockets that contain good designs and merit further consideration. The engineer can then apply traditional tools from optimization and approximation theory to fine-tune preliminary analyses.

Data mining is a key solution approach for such applications, supporting analysis, visualization, and design tasks.[2] It serves a primary role in many domains and a complementary role in others by augmenting traditional techniques from numerical analysis, statistics, and machine learning.

Three important characteristics distinguish the applications studied in this article. First, they are characterized not by an abundance of data, but rather a scarcity of it (owing to the cost and time involved in conducting simulations). Second, the computational scientist has complete control over data acquisition (for example, regions of the design space where he or she can collect data), especially via computer simulations. Finally, significant domain knowledge exists in the form of physical properties such as continuity, correspondence, and locality. Using such information to focus data collection for data mining is thus natural.

This combination of data scarcity plus control over data collection plus the ability to exploit domain knowledge characterizes many important computational science applications. In

1521-9615/02/$17.00 © 2002 IEEE

NAREN RAMAKRISHNAN
Virginia Tech
CHRIS BAILEY-KELLOGG
Purdue University

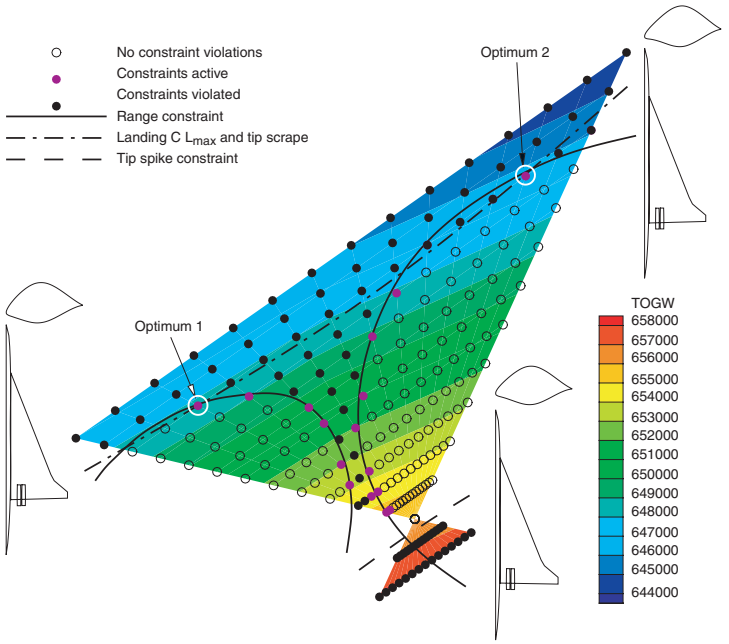

Figure 1. A pocket in an aircraft design space viewed as a slice through three design points. This problem domain involves 29 design variables with 68 constraints. Figure courtesy of Layne T. Watson.

this article, we are interested in the question, "Given a simulation code, knowledge of physical properties, and a data mining goal, at what points should we collect data?" By suitably formulating an objective function and constraints around this question, we can pose it as a problem of minimizing the number of samples needed for data mining.

This article describes focused sampling strategies for mining scientific data. Our approach is based on the Spatial Aggregation Language,[3] which supports construction of data interpretation and control design applications for spatially distributed physical systems in a bottom-up manner. Used as a basis for describing data mining algorithms, SAL programs also help exploit knowledge of physical properties such as continuity and locality in data fields. In conjunction with this process, we introduce a top-down sampling strategy that focuses data collection in only those regions that are deemed most important to support a data mining objective. Together, these processes define a methodology for mining in data-scarce domains. We describe this methodology at a high level and devote the major part of the article to two applications that use it.

Mining in data-scarce domains

Much research focuses on the problem of sampling for targeted data mining activities, such as clustering, finding association rules, and decision tree construction.[4,5] Here, however, we are interested in a general framework or language that expresses data mining operations on data sets and that can help us study the design of data collection and sampling strategies. SAL is such a framework.[3,6]

SAL

As a data mining framework, SAL is based on successive manipulations of data fields by a uniform vocabulary of aggregation, classification, and abstraction operators. Programming in SAL follows a philosophy of building a multilayer hierarchy of aggregations of data. These increasingly abstract data descriptions are built using explicit representations of physical knowledge, expressed as metrics, adjacency relations, and equivalence predicates. This lets a SAL program uncover and exploit structures in physical data.

SAL programs use an imagistic reasoning style.[7] They employ vision-like routines to manipulate multilayer geometric and topological structures in spatially distributed data. SAL adopts a field ontology in which the input is a field mapping from one continuum to another. Multilayer structures arise from continuities in fields at multiple scales. Owing to continuity, fields exhibit regions of uniformity, which can be abstracted as higher-level structures, which in turn exhibit their own continuities. Task-specific domain knowledge describes how to uncover such regions of uniformity, defining metrics for closeness of both field objects and their features. For example, isothermal contours are connected curves of nearby points with equal (or similar enough) temperature.

The identification of structures in a field is a form of data reduction: a relatively information-rich field representation is abstracted into a more concise structural representation (for example, pressure data points into isobar curves or pressure cells, isobar curve segments into troughs). Navigating the mapping from field to abstract description through multiple layers rather than in one giant step allows the construction of more modular programs with more manageable pieces that can use similar processing techniques at different levels of abstraction. The multilevel mapping also lets higher-level layers use the global properties of lower-level objects as local properties. For example, the average temperature in a

region is a global property with respect to the temperature data points but a local property with respect to a more abstract region description. As this article demonstrates, analysis of higher-level structures in such a hierarchy can guide interpretation of lower-level data.

SAL supports structure discovery through a small set of generic operators (parameterized with domain-specific knowledge) on uniform data types. These operators and data types mediate increasingly abstract descriptions of the input data (see Figure 2) to form higher-level abstractions and mine patterns. The primitives in SAL are contiguous regions of space called *spatial objects*, the compounds are collections of spatial objects, and the abstraction mechanisms connect collections at one level of abstraction with single objects at a higher level.

SAL is available as a C++ library, providing access to a large set of data type implementations and operations (download the SAL implementation from www.parc.com/zhao/sal-code.html). In addition, an interpreted interaction environment layered over the library supports rapid prototyping of data mining applications. It lets users inspect data and structures, test the effects of different predicates, and graphically interact with representations of the structures.

To illustrate SAL programming style, consider the task of bundling vectors in a given vector field (for example, wind velocity or temperature gradient) into a set of streamlines (paths through the field following the vector directions). Figure 3 depicts this process; Figure 4 shows the corresponding SAL data mining program. This program has the following steps:

- (a) Establish a field that maps points (locations) to points (vector directions, assumed here to be normalized).
- (b) Localize computation with a neighborhood graph, so that only spatially proximate points are compared.
- (c-f) Use a series of local computations on this representation to find equivalence classes of neighboring vectors with respect to vector direction.
- (g) Redescribe equivalence classes of vectors into more abstract streamline curves.
- (h) Aggregate and classify these curves into groups with similar flow behavior, using the exact same operators but with different metrics (code not shown).

As this example illustrates, SAL provides a vocabulary for expressing the knowledge required (distance and similarity metrics) for uncovering multilevel structures in spatial data sets. Researchers have applied it to applications ranging from decentralized control design[8] to analysis of diffusion-reaction morphogenesis.[9]

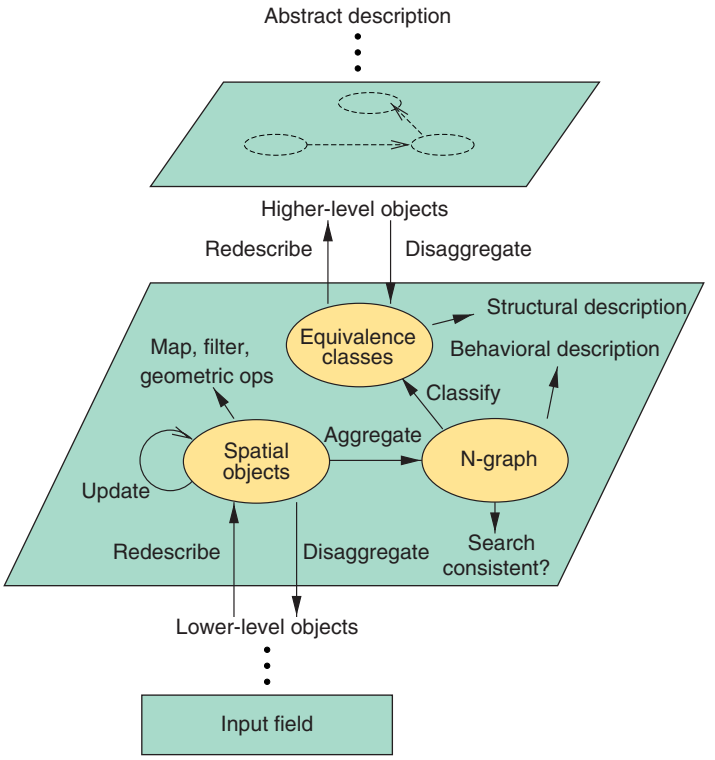

Figure 2. The Spatial Aggregation Language's multilayer spatial aggregates, uncovered by a uniform vocabulary of operators using domain knowledge. We can express several scientific data mining tasks—such as vector field bundling, contour aggregation, correspondence abstraction, clustering, and uncovering regions of uniformity—as multilevel computations with SAL aggregates.

Data collection and sampling

The exploitation of physical properties is a central tenet of SAL because it drives the computation of multilevel spatial aggregates. We can express many important physical properties as SAL computations by suitably defining adjacency relations and aggregation metrics. To extend SAL to data-scarce settings, we present the sampling methodology that Figure 5 outlines.

Once again, understanding the methodology in the context of the vector field bundling application is easy (see Figure 3). Assume that we apply Figure 4's SAL data mining program with a small data set and have navigated up to the highest level

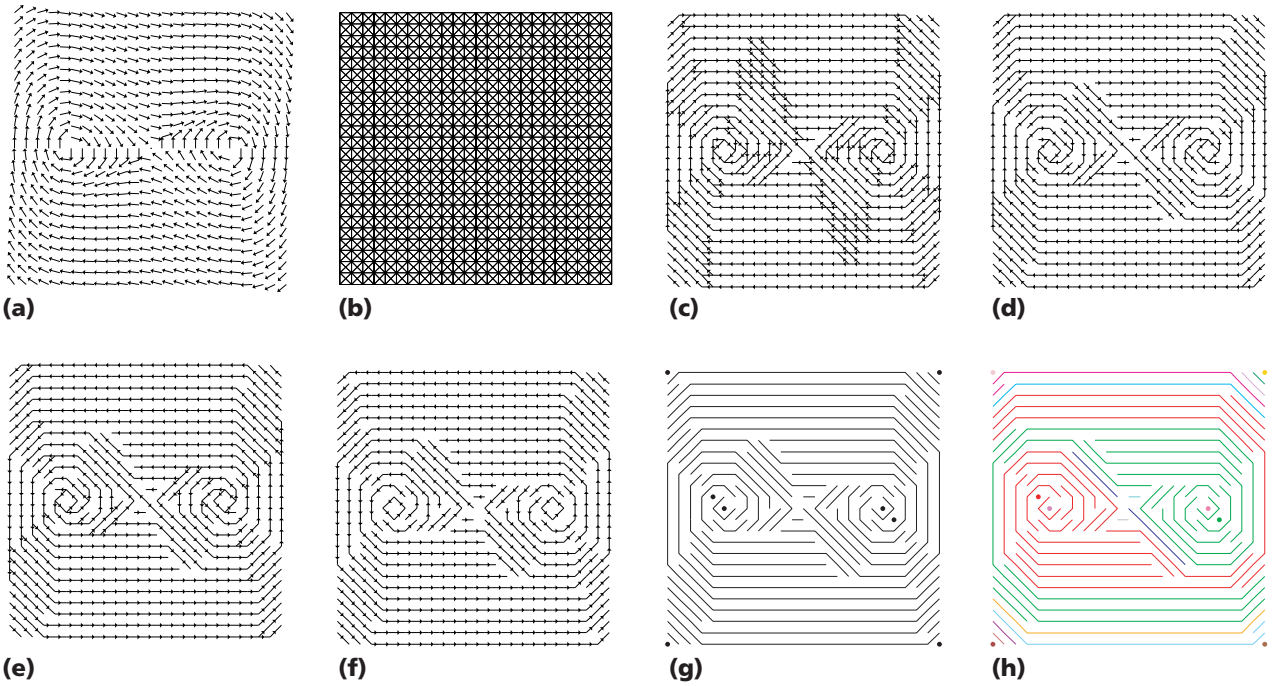

Figure 3. Example steps in a SAL implementation of a vector field analysis application. (a) Input vector field; (b) 8-adjacency neighborhood graph; (c) forward neighbors; (d) best forward neighbors; (e) *N*-graph transposed from best forward neighbors; (f) best backward neighbors; (g) resulting adjacencies redescribed as curves; and (h) higher-level aggregation and classification of curves whose flows converge.

of the hierarchy (streamlines bundled with convergent flows). The SAL program computes different streamline aggregations from a neighborhood graph and chooses one on the basis of how well its curvature matches the direction of the vectors it aggregates. If data is scarce, some of these classification decisions could be ambiguous—multiple streamline aggregations might exist. In such a case, we would want to choose a new data sample that reduces the ambiguity and clarifies what the correct classification should be.

This is the essence of our sampling methodology: using SAL aggregates, we identify an information-theoretic measure (here, ambiguity) that can drive stages of future data collection. For instance, we can summarize the ambiguous streamline classifications as a 2D ambiguity distribution that has a spike for every location where we detected an ambiguity. Ambiguity reduction is a problem of minimizing (or maximizing, as the case may be) a functional involving the computed ambiguity. The functional could be the underlying data field's entropy, as the ambiguity distribution reveals. Such a minimization will lead us to select a data point that clarifies the distribution of streamlines, and hence that more effectively uses data for data mining purposes. This methodology's net effect is that we can capture a particular design's desirability in terms of computations involving SAL aggregates. Thus, sampling is conducted for the express purpose of improving the quality and efficacy of data mining. The data set is updated with the newly collected value, and the process repeats until it meets a desired stopping criteria. We could terminate if the functional is within accepted bounds or when confidence of data mining results does not improve between successive rounds of data collection—in our case, when there is no further ambiguity.

Researchers have studied this idea of sampling to satisfy particular design criteria in various contexts, especially spatial statistics.[10–12] Many of these approaches (including ours) rely on capturing a desirable design's properties in terms of a novel objective function. Our work's distinguishing feature is that it uses spatial information gleaned from a higher level of abstraction to focus data collection at the field or simulation code layer.

Before we proceed, we must note an optional step in our methodology. We could use the newly collected data value to improve a surrogate model, which then generates a dense data field for mining. We would use a surrogate func-

```
// (a) Read vector field.
vect_field = read_point_point_field(infile);
points = domain_space(vect_field);

// (b) Aggregate with 8-adjacency (i.e. within 1.5 units).
point_ngraph = aggregate(points, make_ngraph_near(1.5));

// (c) Compare vector directions with node-neighbor direction.
angle = function (p1, p2) {
     dot(normalize(mean(feature(vect_field, p1), feature(vect_field, p2))),
           normalize(subtract(p2, p1)))
}
forward_ngraph = filter_ngraph(adj in point_ngraph, {
     angle(from(adj), to(adj)) > angle_similarity
})
// (d) Find best forward neighbor, comparing vector direction
// with ngraph edge direction and penalizing for distance.
forward_metric = function (adj) {
     angle(from(adj), to(adj)) - distance_penalty * distance(from(adj),to(adj))
}
best_forward_ngraph = best_neighbors_ngraph(forward_ngraph, forward_metric);

// (e) Find backward neighbors by transposing best forward neighbors.
backward_ngraph = transpose_ngraph(best_forward_ngraph);

// (f) At junctions, keep best backward neighbor using metric
// similar to that for best forward neighbors.
backward_metric = function (adj) {
     angle(to(adj), from(adj)) - distance_penalty*distance(from(adj),to(adj))
}
best_backward_ngraph = best_neighbors_ngraph(backward_ngraph, backward_metric);

// (g) Move to a higher abstraction level by forming equivalence classes
// from remaining groups and redescribing them as curves.
final_ngraph = symmetric_ngraph(best_backward_ngraph, extend=true);
point_classes = classify(points, make_classifier_transitive(final_ngraph));

points_to_curves = redescribe(classes(point_classes),
     make_redescribe_op_path_nline(final_ngraph));
trajs = high_level_objects(points_to_curves);
```

Figure 4. A SAL data mining program for Figure 3's vector field analysis application.

tion in lieu of the real data source to generate sufficient data for mining purposes, which is often more advantageous than working directly with sparse data. Surrogate models are widely used in engineering design, optimization, and response-surface approximations.[13,14]

Example applications

Together, SAL and our focused sampling methodology address the main issues raised in the beginning of this article: SAL's uniform use of fields and abstraction operators lets us exploit prior knowledge in a bottom-up manner. The sampling methodology uses discrepancies as suggested by our knowledge of physical properties in a top-down manner. Continuing these two stages alternatively leads to a closed-loop data mining solution for data-scarce domains (see Figure 5). Let's look at two examples—mining pockets in spatial data and qualitative determination of Jordan forms of matrices—that demonstrate this approach.

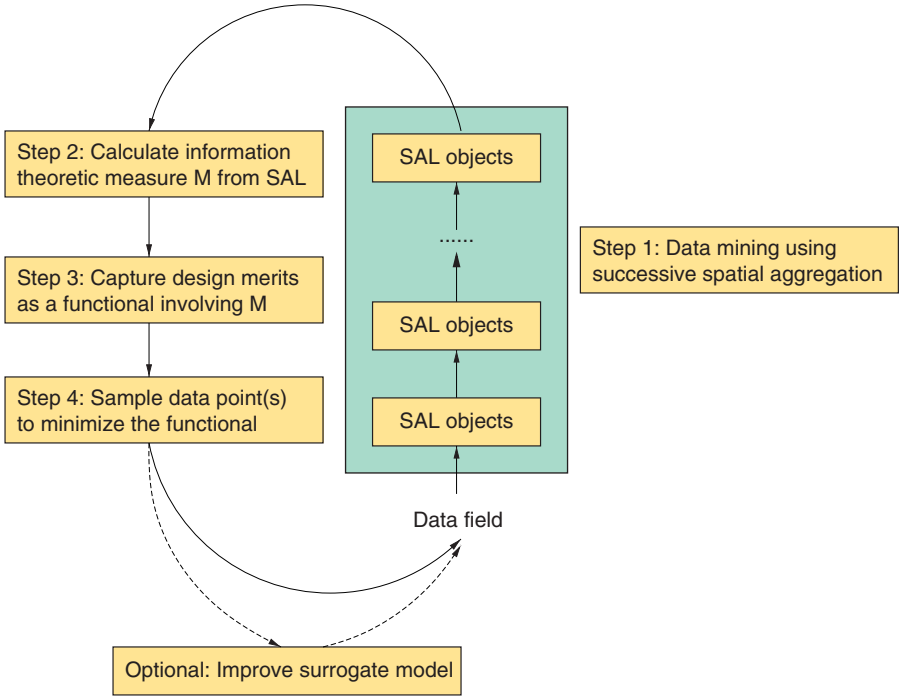

Figure 5. The sampling methodology for SAL mining in data-scarce domains.

Mining pockets in spatial data

Our first application is motivated by the aircraft design problem described in the introduction and illustrates the basic idea of our methodology. Here, we are given a spatial vector field, and we wish to identify pockets underlying the gradient. In a weather map, this might mean identifying pressure troughs. The question is, "Where should data be collected so that we can mine the pockets with high confidence?" We start with a mathematical function that gives rise to pockets in spatial fields. This function will help validate and test our data mining and sampling methodology.

de Boor's function. Carl de Boor invented a pocket function that exploits containment properties of the n-sphere of radius 1 centered at the origin ($\Sigma x_i^2 \leq 1$) with respect to the n-dimensional hypercube defined by $x_i \in [-1, 1]$, $i = 1 \ldots n$. Although the sphere is embedded in the cube, the ratio of the volume of the cube (2^n) to that of the sphere ($\pi^{n/2} / (n/2)!$) grows unboundedly with n. This means that a high-dimensional cube's volume is concentrated in its corners (a counterintuitive notion at first). de Boor exploited this property to design a difficult-to-optimize function that assumes a pocket in each corner of the cube (see Figure 6), just outside the sphere. Formally, we can define it as

$$\alpha(\mathbf{X}) = \cos\left(\sum_{i=1}^{n} 2^i \left(1 + \frac{x_i}{|x_i|}\right)\right) - 2 \quad (1)$$

$$\delta(\mathbf{X}) = \|\mathbf{X} - 0.5\mathbf{I}\| \quad (2)$$

$$p(\mathbf{X}) = \alpha(\mathbf{X})\left(1 - \delta^2(\mathbf{X})(3 - 2\delta(\mathbf{X}))\right) + 1, \quad (3)$$

where $\mathbf{X}$ is the n-dimensional point $(x_1, x_2, \ldots, x_n)$ at which the pocket function p is evaluated, $\mathbf{I}$ is the identity n-vector, and $\| \cdot \|$ is the L_2 norm.

Obviously, p has 2^n pockets (local minima). If n is large (say, 30, which means representing the corners of the n-cube will take more than 500,000 points), naive global optimization algorithms will need an unreasonable number of function evaluations to find the pockets. Our goal for data mining here is to obtain a qualitative indication of the existence, number, and locations of pockets, using low-fidelity models or as few data points as possible. Then, we can use

the results to seed higher-fidelity calculations. This fundamentally differs from DACE (Design and Analysis of Computer Experiments),[12] polynomial response surface approximations,[13] and other approaches in geostatistics where the goal is accuracy of functional prediction at untested data points. Here, we trade accuracy of estimation for the ability to mine pockets.

Surrogate function. In this study, we use the SAL vector field bundling code presented earlier along with a surrogate model as the basis for generating a dense data field. Surrogate theory is an established area in engineering optimization, and we can build a surrogate in several ways. However, the local nature of SAL computations means that we can be selective about our choice of surrogate representation. For example, global, least-squares type approximations are inappropriate because measurements at all locations are equally considered to uncover trends and patterns in a particular region. We advocate the use of *kriging-type interpolators*,[12] which are local modeling methods with roots in Bayesian statistics. Kriging can handle situations with multiple local extrema and can easily exploit anisotropies and trends. Given k observations, the interpolated model gives exact responses at these k sites and estimates values at other sites by minimizing the mean-squared error (MSE), assuming a random data process with zero mean and a known covariance function.

Formally (for two dimensions), we assume the true function p to be the realization of a random process such as

$$p(x, y) = \beta + Z(x, y), \tag{4}$$

where β is typically a uniform random variate, estimated based on the known k values of p, and Z is a correlation function. Kriging then estimates a model p' of the same form, on the basis of the k observations,

$$p'(x_i, y_i) = E(\, p(x_i, y_i) \mid p(x_1, y_1), \ldots, p(x_k, y_k)), \tag{5}$$

and minimizing MSE between p' and p,

$$MSE = E(\, p'(x, y) - p(x, y))^2. \tag{6}$$

A typical choice for Z in p' is $\sigma^2\ R$, where scalar σ^2 is the estimated variance, and correlation matrix R encodes domain-specific constraints and reflects the data's current fidelity. We use an exponential function for entries in R, with two parameters C_1 and C_2:

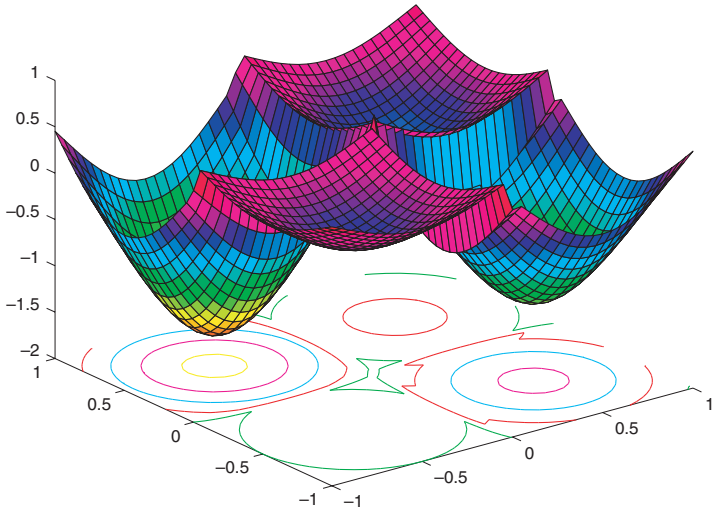

Figure 6. A 2D pocket function.

$$R_{ij} = e^{-C_1\left|x_i - x_j\right|^2 - C_2\left|y_i - y_j\right|^2}. \tag{7}$$

Intuitively, values at closer points should be more highly correlated.

We get the MSE-minimizing estimator by multidimensional optimization (the derivation from Equations 6 and 7 is beyond this article's scope):

$$\max_C \frac{-k}{2}\left(\ln\sigma^2 + \ln|R|\right). \tag{8}$$

This expression satisfies the conditions that there is no error between the model and the true values at the chosen k sites and that all variability in the model arises from the design of Z. Gradient descent or pattern search methods often perform the multidimensional optimization.[12]

Data mining and sampling methodology. The bottom-up computation of SAL aggregates from the surrogate model's outputs will possibly lead to some ambiguous streamline classifications, as we discussed earlier. Ambiguity can reflect the desirability of acquiring data at or near a specified point—to clarify the correct classification and to serve as a mathematical criterion of information content. We can use information about ambiguity to drive data collection in several ways. In this study, we express the ambiguities as a distribution describing the number of possible good neighbors. This ambiguity distri-

bution provides a novel mechanism to include qualitative information; streamlines that agree will generally contribute less to data mining, for information purposes. We thus define the information-theoretic measure M (see Figure 5) to be the ambiguity distribution $\wp$.

We define the functional as the posterior entropy $E(-\log d)$, where d is the conditional density of $\wp$ over the design space not covered by the current data values. By a reduction argument, minimizing this posterior entropy can be shown to maximize the prior entropy over the sampled design space.[12] In turn, this maximizes the amount of information obtained from an experiment (additional data collection). Moreover, we also incorporate $\wp$ as an indicator covariance term in our surrogate model, which is a conventional method for including qualitative information in an interpolatory model.[11]

Experimental results. Our initial experimental configuration used a face-centered design (four points, in the 2D case). A surrogate model by kriging interpolation then generated data on a 41^n-point grid. We used de Boor's function as the source for data values; we also employed pseudorandom perturbations of it that shift the pockets from the corners somewhat unpredictably.[15] In total, we experimented with 200 perturbed variations of the 2D and 3D pocket functions. For each of these cases, we organized data collection in rounds of one extra sample each (to minimize the functional). We recorded the number of samples SAL needed to mine all the pockets and compared our results with those obtained from a pure DACE–kriging approach. In other words, we used the DACE methodology to suggest new locations for data collection and determined how those choices fared with respect to mining the pockets.

Figure 7 shows the distributions of the total number of data samples required to mine the four pockets for the 2D case. We mined the 2D pockets with three to 11 additional samples, whereas the conventional kriging approach required 13 to 19 additional samples. The results were more striking in the 3D case: at most 42 additional samples for focused sampling and up to 151 points for conventional kriging. This shows that our focused sampling methodology performs 40 to 75 percent better than sampling by conventional kriging.

Figure 8a describes a 2D design involving only seven total data points that can mine the four pockets. Counterintuitively, no additional sample is required in the lower-left quadrant. Although this will lead to a highly suboptimal design (from the traditional viewpoint of minimizing variance in predicted values), it is nevertheless an appropriate design for data mining purposes. In particular, this means that neighborhood calculations involving the other three quadrants are enough to uncover the pocket in the fourth quadrant. Because the kriging interpolator uses local modeling and because pockets in 2D effectively occupy the quadrants, obtaining measurements at ambiguous locations captures each dip's relatively narrow regime, which in turn helps distinguish the pocket in the neighboring quadrant. Achieving this effect is hard without exploiting knowledge of physical properties—in this case, locality of the dips.

Qualitative Jordan form determination

In our second application, we use our methodology to identify a given matrix's most probable Jordan form. This is a good application for data mining because the Jordan form's direct computation leads to a numerically unstable algorithm.

Jordan forms. A matrix $\mathcal{A}$ (real or complex) that has r independent eigenvectors has a Jordan form consisting of r blocks. Each of these blocks is an upper triangular matrix that is associated with one of the eigenvectors of $\mathcal{A}$ and whose size describes the corresponding eigenvalue's multiplicity. For the given matrix $\mathcal{A}$, the diagonalization thus posits a nonsingular matrix $\mathcal{B}$ such that

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{B} = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_r \end{bmatrix}, \tag{9}$$

where

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & 1 & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \tag{10}$$

and λ_i is the eigenvalue revealed by the ith Jordan block (J_i). The Jordan form is most easily explained by looking at how eigenvectors are distributed for a given eigenvalue. Consider, for example, the matrix

$$\begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & -1 & 3 \end{bmatrix}, \tag{11}$$

which has eigenvalues at 1, 1, and 2. This matrix has only two eigenvectors, as revealed by the two-block structure of its Jordan form:

$$\left[\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 2 \end{array}\right]. \quad (12)$$

The Jordan form shows that there is one eigenvalue (1) of multiplicity 2 and one eigenvalue (2) of multiplicity 1. We say that the matrix has the Jordan structure given by $(1)^2\ (2)^1$. In contrast, the matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

has the same eigenvalues but a three-block Jordan structure:

$$\left[\begin{array}{c|c|c} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 2 \end{array}\right]. \quad (14)$$

This is because there are three independent eigenvectors (the unit vectors, actually). The diagonalizing matrix is thus the identity matrix, and the Jordan form has three permutations.

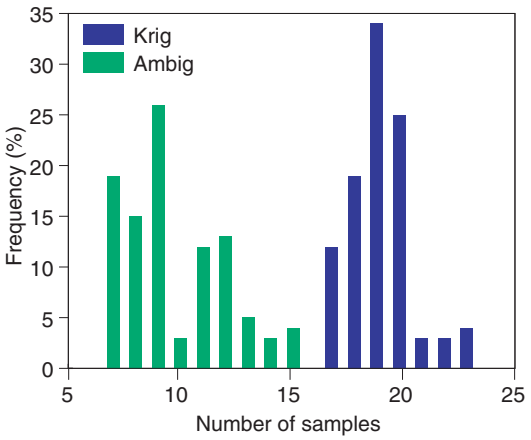

Figure 7. Pocket-finding results for the 2D example show that focused sampling using a measure of ambiguity always requires fewer total samples (7 to 15) than conventional kriging (17 to 23).

The Jordan structure is therefore given by $(1)^1\ (1)^1\ (2)^1$. These two examples show that a given eigenvalue's multiplicity could be distributed across one, many, or all Jordan blocks. Correlating the eigenvalue with the block structure is an important problem in numerical analysis.

The typical approach to computing the Jordan form is to follow the structure's staircase pattern and perform rank determinations in conjunction with ascertaining the eigenvalues. One

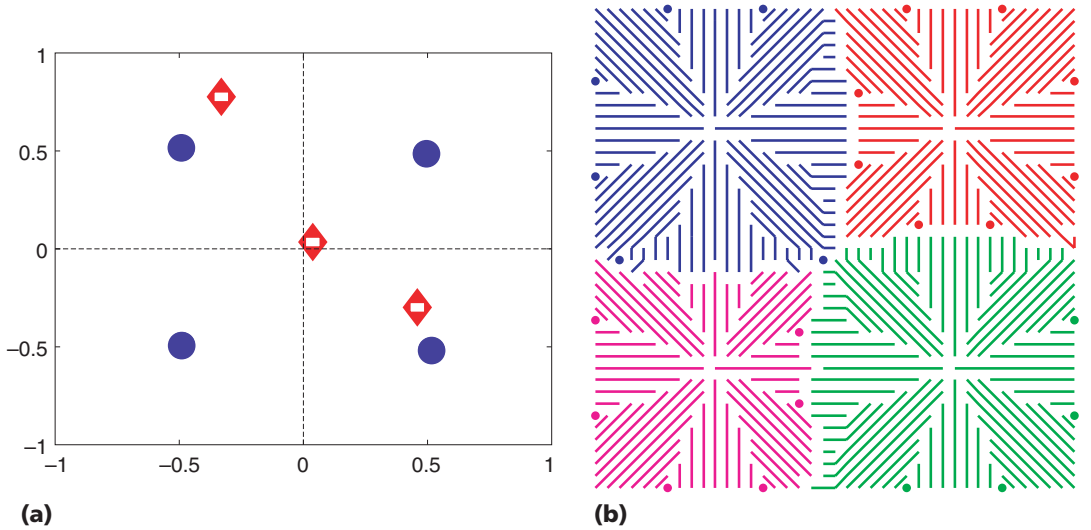

Figure 8. Mining pockets in 2D from only seven sample points. (a) The chosen sample locations: four initial face-centered samples (blue circles) plus three samples selected by our methodology (red diamonds). No additional sample is required in the lower-left quadrant. (b) SAL structures in the surrogate model data, confirming the existence of four pockets.

of the more serious caveats with such an approach involves mistaking an eigenvalue of multiplicity greater than 1 for multiple eigenvalues.[16] In Equation 11, this might lead to inferring that the Jordan form has three blocks. The extra care needed to safeguard staircase algorithms usually involves more complexity than the original computation to be performed. The ill-conditioned nature of this computation has thus traditionally prompted numerical analysts to favor other, more stable, decompositions.

Qualitative assessment of Jordan forms. A recent development has been the acceptance of a qualitative approach to Jordan structure determination, proposed by Françoise Chaitin-Chatelin and Valerie Frayssé.[17] This approach does not use the staircase idea. Instead, it exploits a semantics of eigenvalue perturbations to infer multiplicity, which leads to a geometrically intuitive algorithm that we can implement using SAL.

Consider a matrix that has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ with multiplicities $\rho_1, \rho_2, \ldots, \rho_n$. Any attempt at finding the eigenvalues (for example, determining the roots of the characteristic polynomial) is intrinsically subject to the numerical analysis dogma: the problem being solved will actually be a perturbed version of the original problem. This allows the expression of the computed eigenvalues in terms of perturbations on the actual eigenvalues. The computed eigenvalue corresponding to any λ_k will be distributed on the complex plane as

$$\lambda_k + |\Delta|^{\frac{1}{\rho_k}} e^{\frac{i\phi}{\rho_k}}, \tag{15}$$

where the phase ϕ of the perturbation Δ ranges over $\{2\pi, 4\pi, \ldots, 2\rho_k \pi\}$ if Δ is positive and over $\{3\pi, 5\pi, \ldots, 2(\rho_k + 1)\pi\}$ if Δ is negative. Chaitin-Chatelin and Frayssé superimposed numerous such perturbed calculations graphically so that the aggregate picture reveals the ρ_k of the eigenvalue λ_k.[17] The phase variations imply that the computed eigenvalues will lie on a regular polygon's vertices—centered on the actual eigenvalue—where the number of sides is two times the multiplicity of the considered eigenvalue. This takes into account both positive and negative Δ. Because Δ influences the polygon's diameter, iterating this process over many Δ will lead to a "sticks" depiction of the Jordan form.

To illustrate, we choose a matrix whose computations will be more prone to finite precision errors. Perturbations on the 8×8 Brunet matrix with Jordan structure $(-1)^1\ (-2)^1\ (7)^3\ (7)^3$ induce the superimposed structures in Figure 9.[17] Figure 9a depicts normwise relative perturbations in the scale of $[2^{-50}, 2^{-40}]$. The six sticks around the eigenvalue at 7 clearly reveal that its Jordan block is of size 3. The other Jordan block, also centered at 7, is revealed if we conduct our exploration at a finer perturbation level. Figure 9b reveals the second Jordan block using perturbations in the range $[2^{-53}, 2^{-50}]$. The noise in both pictures is a consequence of having two Jordan blocks with the same size and a "ring" phenomenon studied elsewhere.[18] We do not attempt to capture these effects in this article.

Data mining and sampling methodology. For this study, we collect data by random normwise perturbations in a given region, and a SAL program determines multiplicity by detecting symmetry correspondence in the samples. The first aggregation level collects a given perturbation's samples into triangles. The second aggregation level finds congruent triangles via geometric hashing[19] and uses congruence to establish a correspondence relation among triangle vertices. This relation is then abstracted into a rotation about a point (the eigenvalue) and evaluated for whether each point rotates onto another and whether matches define regular polygons. A third level then compares rotations across different perturbations, revisiting perturbations or choosing new ones to disambiguate (see Figure 10).

The end result of this analysis is a confidence measure on models of possible Jordan forms. Each model is defined by its estimate of λ and ρ (we work in one region at a time). The measure M is the joint probability distribution over the space of λ and ρ.

Experimental results. Because our Jordan form computation treats multiple perturbations irrespective of level as independent estimates of eigenstructure, the idea of sampling here is not where to collect, but how much to collect. The goal of data mining is hence to improve our confidence in model evaluation.

We organized data collection into rounds of six to eight samples each, varied a tolerance parameter for triangle congruence from 0.1 to 0.5 (effectively increasing the number of models posited), and determined the number of rounds needed to determine the Jordan form. As test cases, we used the set of matrices Chaitin-Chatelin and Frayssé studied.[17] On average, our

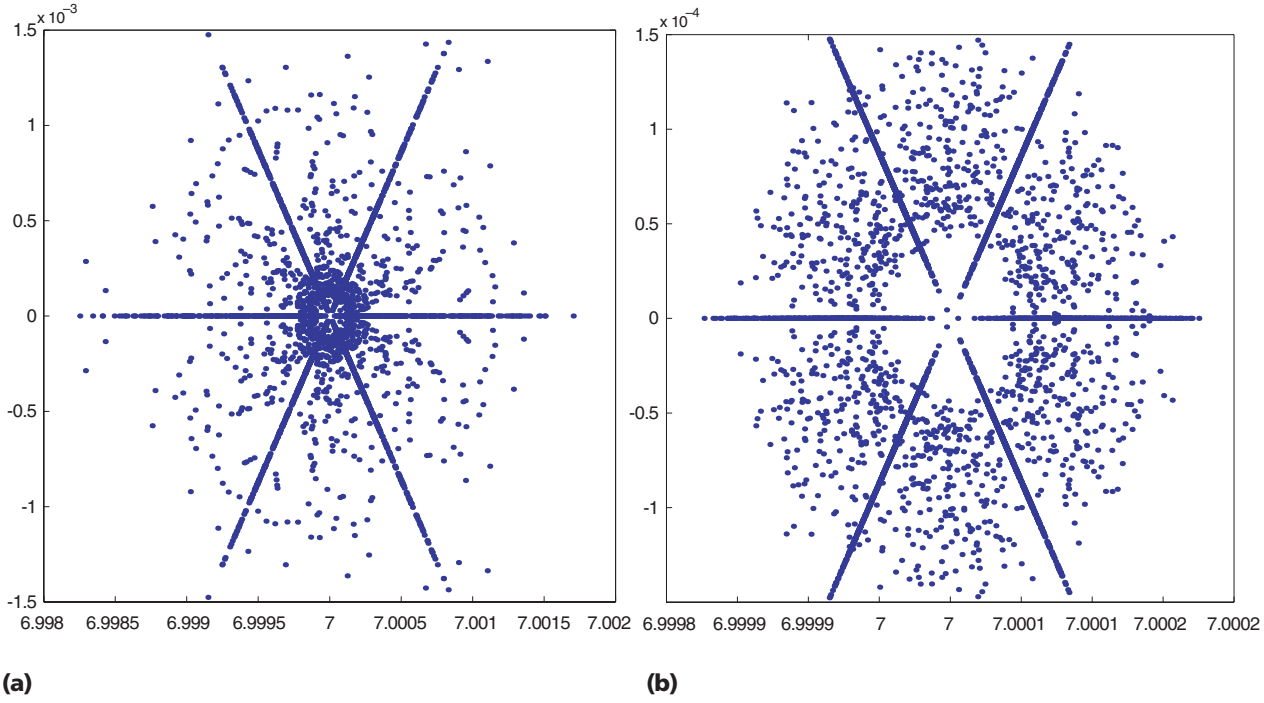

Figure 9. Superimposed spectra for assessing the Jordan form of the Brunet matrix. We see two Jordan blocks of multiplicity 3 for eigenvalue 7, at (a) coarse and (b) fine perturbation levels.

focused sampling approach required one round of data collection at a tolerance of 0.1 and up to 2.7 rounds at 0.5. Even with a large number of models posited, additional data quickly weeded out bad models. Figure 10 demonstrates this mechanism on the Brunet matrix discussed earlier for two sets of sample points. To the best of our knowledge, this is the only known focused sampling methodology for this domain; we are unable to present any comparisons. However, by harnessing domain knowledge about correspondences, we have arrived at an intelligent sampling methodology that resembles what a human would get from visual inspection.

Our methodology for mining in data-scarce domains has several intrinsic benefits. First, it is based on a uniform vocabulary of operators that a rich diversity of applications can exploit. Second, it demonstrates a novel factorization to the problem of mining when data is scarce—namely, formulating an experiment design methodology to clarify, disambiguate, and improve confidences in higher-level aggregates of data. This lets us bridge qualitative and quantitative information in a unified framework. Third, our methodology can coexist with more traditional approaches to problem solving (numerical analysis and optimization); it is not meant to be a replacement or a contrasting approach.

The methodology makes several intrinsic assumptions that we only briefly mention here. Both our applications have been such that the cause, formation, and effect of the relevant physical properties are well understood. This is precisely what lets us act decisively on the basis of higher-level information from SAL aggregates, through the measure M. It also assumes that the problems the mining algorithm will encounter are the same as the problems for which it was designed. This is an inheritance from Bayesian inductive inference and leads to fundamental limitations on what we can do in such a setting. For instance, if new data does not help clarify an ambiguity, does the fault lie with the model or the data? We can summarize this problem by saying that the approach requires strong a priori information about what is possible and what is not.

Nevertheless, by advocating targeted use of domain-specific knowledge and aiding qualitative model selection, our methodology is more efficient at determining high-level models from empirical data. Together, SAL and our informa-

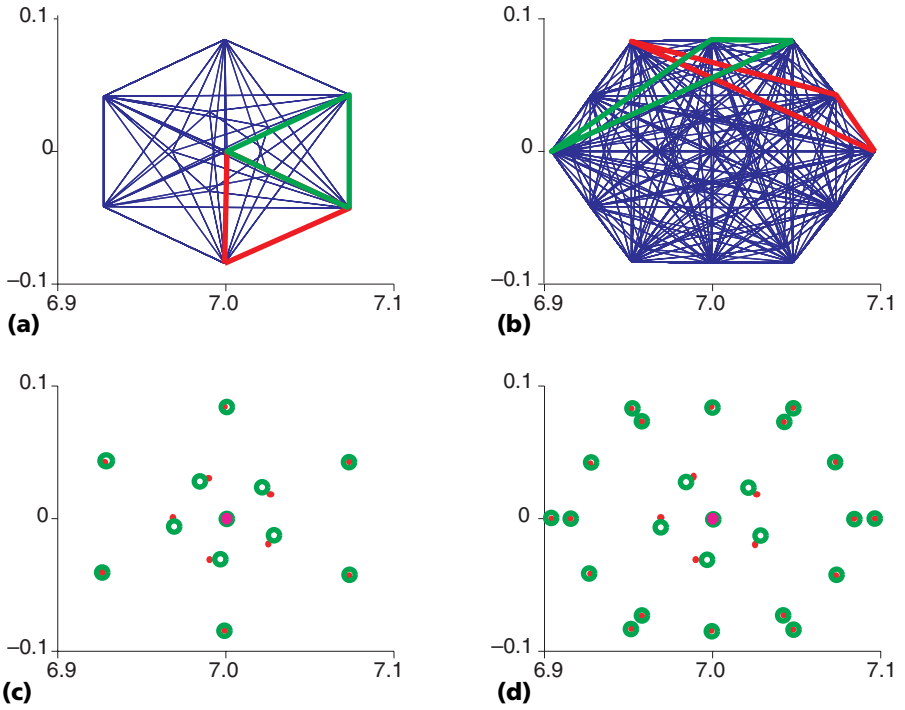

Figure 10. Mining Jordan forms from a small ((a) and (c)) sample set and a larger ((b) and (d)) sample set. Sections (a) and (b) depict selected approximately congruent triangles; (c) and (d) evaluate the correspondence of the original sample points with their images under a rotation aligning the triangles.

tion-theoretic measure M encapsulate knowledge about physical properties, which is what makes our methodology viable for data mining. In the future, we aim to characterize more formally the particular forms of domain knowledge that help overcome sparsity and noise in scientific data sets.

We could also extend our framework to take into account the expense of data samples. If the cost of data collection is nonuniform across the domain, then including this in the design of our functional will let us trade the cost of gathering information with the expected improvement in problem-solving performance. This area of data mining is called *active learning*.

Data mining can sometimes be a controversial term in a discipline that is used to mathematical rigor; this is because it is often used synonymously with "lack of a hypothesis or theory." This need not be the case. Data mining can indeed be sensitive to knowledge about the domain, especially physical properties of the kind we have harnessed here. As data mining applications become more prevalent in science, the need to incorporate a priori domain knowledge will become even more important.

Acknowledgments

This work is supported in part by US National Science Foundation grants EIA-9984317 and EIA-0103660. We thank John R. Rice, Feng Zhao, and Layne T. Watson for their helpful comments.

References

1. A. Goel et al., "VizCraft: A Problem-Solving Environment for Aircraft Configuration Design," *Computing in Science & Eng.*, vol. 3, no. 1, Jan./Feb. 2001, pp. 56–66.
2. N. Ramakrishnan and A.Y. Grama, "Mining Scientific Data," *Advances in Computers*, vol. 55, Sept. 2001, pp. 119–169.
3. C. Bailey-Kellogg, F. Zhao, and K. Yip, "Spatial Aggregation: Language and Applications," *Proc. 13th Nat'l Conf. Artificial Intelligence* (AAAI 96), AAAI Press, Menlo Park, Calif., 1996, pp. 517–522.
4. V. Ganti, J. Gehrke, and R. Ramakrishnan, "Mining Very Large Databases," *Computer*, vol. 32, no. 8, Aug. 1999, pp. 38–45.
5. J. Kivinen and H. Mannila, "The Use of Sampling in Knowledge Discovery," *Proc. 13th ACM Symp. Principles of Database Systems*, ACM Press, New York, 1994, pp. 77–85.
6. K.M. Yip and F. Zhao, "Spatial Aggregation: Theory and Applications," *J. Artificial Intelligence Research*, vol. 5, 1996, pp. 1–26.
7. K.M. Yip, F. Zhao, and E. Sacks, "Imagistic Reasoning," *ACM Computing Surveys*, vol. 27, no. 3, Sept. 1995, pp. 363–365.
8. C. Bailey-Kellogg and F. Zhao, "Influence-Based Model Decomposition for Reasoning about Spatially Distributed Physical Systems," *Artificial Intelligence*, vol. 130, no. 2, Aug. 2001, pp. 125–166.

9. I. Ordóñez and F. Zhao, "Spatio-Temporal Aggregation with Applications to Analysis of Diffusion-Reaction Phenomena," *Proc. 17th Nat'l Conf. Artificial Intelligence* (AAAI 00), AAAI Press, Menlo Park, Calif., 2000, pp. 517–523.
10. R.G. Easterling, "Comment on 'Design and Analysis of Computer Experiments,'" *Statistical Science*, vol. 4, no. 4, Nov. 1989, pp. 425–427.
11. A. Journel, "Constrainted Interpolation and Qualitative Information: The Soft Kriging Approach," *Mathematical Geology*, vol. 18, no. 2, Nov. 1986, pp. 269–286.
12. J. Sacks et al., "Design and Analysis of Computer Experiments," *Statistical Science*, vol. 4, no. 4, Nov. 1989, pp. 409–423.
13. D.L. Knill et al., "Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design," *J. Aircraft*, vol. 36, no. 1, Jan. 1999, pp. 75–86.
14. R.H. Myers and D.C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York, 2002.
15. C. Bailey-Kellogg and N. Ramakrishnan, "Ambiguity-Directed Sampling for Qualitative Analysis of Sparse Data from Spatially Distributed Physical Systems," *Proc. 17th Int'l Joint Conf. Artificial Intelligence* (IJCAI 01), Morgan Kaufmann, San Francisco, 2001, pp. 43–50.
16. A. Edelman and Y. Ma, "Staircase Failures Explained by Orthogonal Versal Forms," *SIAM J. Matrix Analysis and Applications*, vol. 21, no. 3, 2000, pp. 1004–1025.
17. F. Chaitin-Chatelin and V. Frayssé, *Lectures on Finite Precision Computations*, Soc. For Industrial and Applied Mathematics, Monographs, Philadelphia, 1996.
18. A. Edelman and Y. Ma, "Non-Generic Eigenvalue Perturbations of Jordan Blocks," *Linear Algebra & Applications*, vol. 273, nos. 1–3, Apr. 1998, pp. 45–63.
19. Y. Lamdan and H. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proc. 2nd Int'l Conf. Computer Vision* (ICCV), IEEE CS Press, Los Alamitos, Calif., 1988, pp. 238–249.

Naren Ramakrishnan is an assistant professor of computer science at Virginia Tech. His research interests include problem-solving environments, mining scientific data, and personalization. He received his PhD in computer sciences from Purdue University. Contact him at the Dept. of Computer Science, 660 McBryde Hall, Virginia Tech, Blacksburg, VA 24061; naren@cs.vt.edu.

Chris Bailey-Kellogg is an assistant professor of computer sciences at Purdue. His research combines geometric, symbolic, and numeric approaches for data analysis and experiment planning in scientific and engineering domains. He received his BS and MS in electrical engineering and computer science from MIT and his PhD in computer and information science from Ohio State University. Contact him at the Dept. of Computer Sciences, 1398 Computer Science Bldg., Purdue Univ., West Lafayette, IN 47907; cbk@cs.purdue.edu.

For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

AMERICAN INSTITUTE OF PHYSICS

Member Societies
American Physical Society
Optical Society of America
Acoustical Society of America
The Society of Rheology
American Association of Physics Teachers
American Crystallographic Association
American Astronomical Society
American Association of Physicists in Medicine
AVS
American Geophysical Union
Other Member Organizations
Sigma Pi Sigma, Physics Honor Society
Society of Physics Students
Corporate Associates

The American Institute of Physics is a not-for-profit membership corporation chartered in New York State in 1931 for the purpose of promoting the advancement and diffusion of the knowledge of physics and its application to human welfare. Leading societies in the fields of physics, astronomy, and related sciences are its members.

The Institute publishes its own scientific journals as well as those of its Member Societies; provides abstracting and indexing services; provides online database services; disseminates reliable information on physics to the public; collects and analyzes statistics on the profession and on physics education; encourages and assists in the documentation and study of the history and philosophy of physics; cooperates with other organizations on educational projects at all levels; and collects and analyzes information on Federal programs and budgets.

The scientists represented by the Institute through its Member Societies number approximately 120,000. In addition, approximately 5,400 students in over 600 colleges and universities are members of the Institute's Society of Physics Students, which includes the honor society Sigma Pi Sigma. Industry is represented through 47 Corporate Associates members.

Governing Board*
John A. Armstrong, (Chair), *Marc H. Brodsky* (Executive Director), *Benjamin B. Snavely* (Secretary), Martin Blume (APS), William F. I. Brinkman (APS), Judy R. Franz (APS),Donald R. Hamann (APS), Myriam P. Sarachik (APS), *Thomas J. McIlrath* (APS), George H. Trilling (APS), *Michael D. Duncan* (OSA), Ivan P. Kaminow (OSA), Anthony M. Johnson (OSA), Elizabeth A. Rogan (OSA), Anthony A. Atchley (ASA), *Lawrence A. Crum* (ASA), Charles E. Schmid (ASA), Arthur B. Metzner (SOR), Christopher J. Chiaverina (AAPT), Charles H. Holbrow (AAPT), John Hubisz (AAPT), *Bernard V. Khoury* (AAPT), Charlotte Lowe-Ma (ACA), S. Narasinga Rao (ACA), Leonard V. Kuhi (AAS), Arlo U. Landolt (AAS), Robert W. Milkey (AAS), *James B. Smathers* (AAPM), Christopher H. Marshall (AAPM), *Rudolf Ludeke* (AVS), N. Rey Whetten (AVS), Dawn A. Bonnell (AVS), James L. Burch (AGU), Robert E. Dickinson (AGU), Jeffrey J. Park (AGU), Judy C. Holoviak (AGU), *Louis J. Lanzerotti* (AGU), Fred Spilhaus (AGU), Brian Clark (2002) MAL, Frank L. Huband (MAL)
**Executive Committee members are printed in italics.*

Management Committee
Marc H. Brodsky, Executive Director and CEO; Richard Baccante, Treasurer and CFO; Theresa C. Braun, Vice President, Human Resources; James H. Stith, Vice President, Physics Resources; Darlene A. Walters, Senior Vice President, Publishing; Benjamin B. Snavely, Secretary

Subscriber Services
AIP subscriptions, renewals, address changes, and single-copy orders should be addressed to Circulation and Fulfillment Division, American Institute of Physics, 1NO1, 2 Huntington Quadrangle, Melville, NY 11747-4502. Tel. (800) 344-6902; e-mail subs@aip.org. Allow at least six weeks' advance notice. For address changes please send both old and new addresses, and, if possible, include an address label from the mailing wrapper of a recent issue.