Active Data Mining of Correspondence for Qualitative Assessment of Scientific Computations

Chris Bailey-Kellogg
Purdue Computer Science Dept.
250 N. Univ. St.
West Lafayette, IN 47907
cbk@cs.purdue.edu

Naren Ramakrishnan
2160L Torgersen Hall
Department of Computer Science
Virginia Tech, VA 24061
naren@cs.vt.edu

Abstract

Active data mining constructs and evaluates possible models explaining a dataset, and reasons about the cost and impact of additional samples on refining and selecting among the models. It is particularly appropriate for applications characterized by expensive data collection, from either experiment or simulation. This paper develops an active mining mechanism based on a multi-level, qualitative analysis of correspondence. Correspondence operators presented here leverage domain knowledge to establish relationships among objects, evaluate implications for model selection, and leverage identified weaknesses to focus additional data collection. The utility of the qualitative framework is demonstrated in two scientific computing applications — matrix spectral portrait analysis and graphical assessment of Jordan forms of matrices. Results show that the mechanism efficiently samples computational experiments and successfully uncovers high-level properties of data. The framework helps overcome noise and sparsity by leveraging domain knowledge to detect mutually reinforcing interpretations of spatial data.

Introduction

Active data mining is concerned with the problem of integrating data collection, experiment design, and data mining, with the end-goal of making better use of data for data mining purposes. In applications where we have control over the data acquisition process, we would like to select samples from those locations that present the greatest benefit for identifying high-level structures and models underlying the data. Active mining is especially crucial in sparse data contexts, where each data point is costly (e.g., in terms of time, computational effort) and it is beneficial to make judicious choices of locations to sample.

This paper develops an active mining mechanism based on a novel, multi-level, qualitative analysis of correspondence. We develop our mechanism in the context of qualitative assessment of scientific computations (Chaitin-Chatelin & Frayssé 1996), where the goal is to empirically characterize problem characteristics (e.g., matrix sensitivity) and algorithm performance (e.g., convergence) by a data-driven strategy. This approach is becoming preferred in applications where domain knowledge is imperfect and where theory-driven approaches are inadequate. For instance, when solving linear systems associated with finite-difference discretization of elliptic partial differential equations (PDEs), there is little mathematical theory to guide a choice between, say, a direct solver and an iterative Krylov solver plus preconditioner. A qualitative assessment approach is to parameterize a suitable family of problems, and mine a database of PDE “solves” to gain insight into the likely relative performance of these two approaches (Ramakrishnan & Ribbens 2000).

Many tasks in scientific computing involve assessing the eigenstructure of a given matrix. Eigenstructure helps characterize the stability, sensitivity, and accuracy of numerical methods as well as the fundamental tractability of problems. Recently, the *spectral portrait* (see Fig. 1) has emerged as a popular tool for graphically visualizing eigenstructure. A spectral portait characterizes how the eigenvalues of a matrix change as perturbations (e.g. due to numerical error) are introduced in computations involving the matrix. Level curves in a spectral portait correspond to perturbation magnitudes, and the region enclosed by a level curve contains all possible eigenvalues that are equivalent with respect to perturbations of a given magnitude. Analysis of level curves with respect to a class of perturbations reveals information about the matrix (e.g. nonnormality and defective eigenvalues) and the effects of different algorithms and numerical approximations.

The goal thus is to determine *high-level properties* by analyzing data from *low-level computational experiments*. We focus here on the particular case where such properties are extracted from *graphical representations* and where it is necessary to minimize the computational experiments performed (owing to the cost and complexity of conducting them). We pose the extraction of high-level properties as a model selection problem and pursue an *active data mining* approach, where reasoning about a set of models drives additional data collection in order to refine and discriminate them. Each step in our framework is parameterized by domain knowledge of properties such as locality, similarity, and correspondence. Therefore the framework is generic both with respect to a variety of problems in scientific computing (two case studies are presented here), and to problems in other domains (e.g. connections to weather data analysis are discussed). Furthermore, the approach leads to efficient and explainable data collection motivated directly by the need to disambiguate among high-level models.

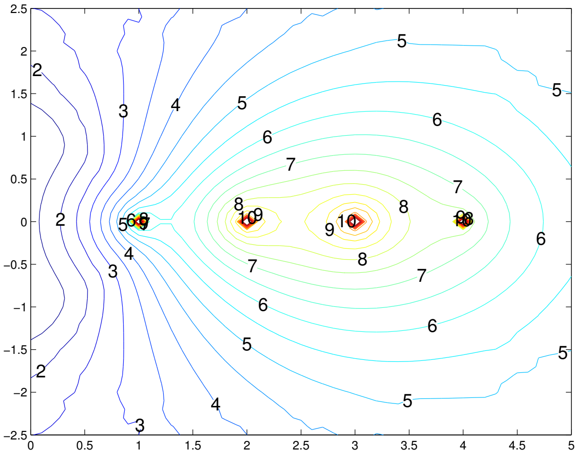

Figure 1: An example spectral portrait, for a matrix with eigenvalues at 1, 2, 3, and 4. The portrait graphically illustrates, in the complex plane, the eigenstructure of a given matrix. Qualitative properties of the portrait correspond to important characteristics of the underlying problem that must be considered when selecting and applying numerical algorithms. For instance, a level curve in the portrait shown here bounds a set of eigenvalues (points) that are indistinguishable with respect to matrix perturbations at a particular magnitude. Perturbation levels increase going outward from singularities at the (unperturbed) eigenvalues, and a curve surrounding multiple eigenvalues indicates a level of precision (e.g. 10^{-6} for the curve labeled "6") beyond which those eigenvalues cannot be distinguished.

Qualitative Analysis of Spatial Data

Correspondence is a ubiquitous concept in the interpretation of spatial datasets and plays a central role in our qualitative analysis. Correspondence establishes analogy, indicating objects that play similar roles with respect to some context. For example, correspondence between template and image features supports object recognition, correspondence among isobars in a weather map aids identification of pressure troughs and ridges, and, as shown in this paper, correspondence and lack thereof among level curves in datasets like Fig. 1 supports characterization of matrix properties for scientific computing applications.

Our mechanism for qualitative analysis of correspondence is based on the Spatial Aggregation Language (SAL) (Bailey-Kellogg, Zhao, & Yip 1996; Yip & Zhao 1996) and the ambiguity-directed sampling framework (Bailey-Kellogg & Ramakrishnan 2001). SAL programs apply a set of uniform operators and data types (Fig. 2) in order to extract multi-layer geometric and topological representations of spatial data. These operators utilize domain knowledge of physical properties such as continuity and locality, specified as metrics, adjacency relations, and equivalence predicates, to uncover regions of uniformity in spatially distributed data. Ambiguity-directed sampling is an active data mining mechanism, focusing data collection so as to clarify difficult choice points in an aggregation hierarchy. It seeks to maximize information content while minimizing the number and expense of data samples.

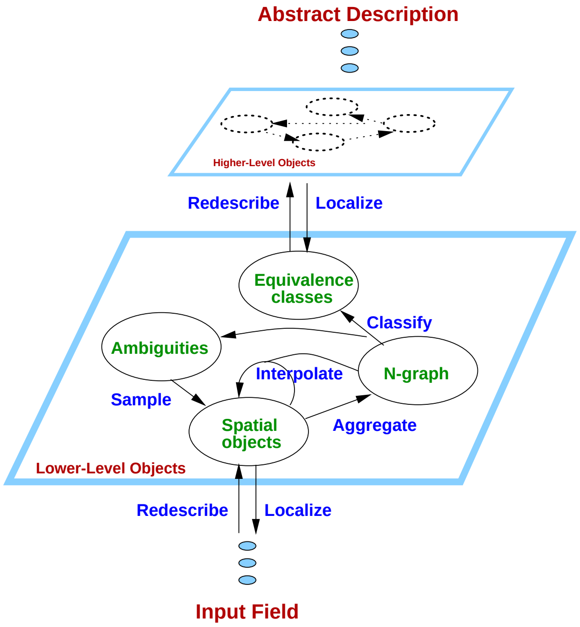

Figure 2: The spatial aggregation language provides a uniform vocabulary of operators utilizing domain knowledge to build multi-layer structural descriptions of spatial data.

As an example of aggregation, consider the construction of level curves in the spectral portrait in Fig. 1. The key steps in such an analysis (after (Huang & Zhao 1999)) are:

- The *input field* maps a set of discrete sample locations (e.g. on a uniform grid) to perturbation levels, representing allowable imprecision before a point becomes indistinguishable from an eigenvalue (computational details are discussed later).
- *Aggregate* points in a *neighborhood graph*, localizing computation to spatially proximate points, e.g. in a Delaunay triangulation or regular grid.
- *Interpolate* values at new locations from values at nearby samples according to the field, in this case determining locations of points with perturbation level belonging to a discrete set (e.g. 10^{-1}, 10^{-2}, $\ldots$).
- *Classify* neighboring similar-enough objects into equivalence classes with an *equivalence predicate*, in this case testing equality of field value.[1]
- *Redescribe* each equivalence class of lower-level objects as a single higher-level object, in this case abstracting connected points into curves. The curve can be represented more compactly and abstractly (e.g. with a spline) than its set of sample points.

[1] While an implementation such as marching squares might combine interpolation, aggregation, and classification, we view them as conceptually distinct operations.

As a consequence of redescription, the next aggregation level can treat curves as first-class objects, and aggregate, classify, and redescribe them, for example to find curves nested around a single eigenvalue. This higher-level process uses the same operators, but with different parameters specifying locality, equivalence, and abstraction. Ambiguity arises when, for example, not enough sample points are available to be confident in a curve's location, or in the separation of two curves. Ambiguity-directed sampling then optimizes selection of new locations (e.g. near the ambiguity) for data collection in order to clarify the decision-making.

As this example illustrates, SAL and ambiguity-directed sampling provide a suitable *vocabulary* (e.g. distance and similarity metrics) and *mechanism* (bottom-up aggregation and top-down sampling) to uncover multi-level structures in spatial data sets. Successful applications include decentralized control design (Bailey-Kellogg & Zhao 1999; 2001) weather data analysis (Huang & Zhao 1999), analysis of diffusion-reaction morphogenesis (Ordóñez & Zhao 2000), and identification of pockets underlying gradient fields and decomposition of a field based on control influences (Bailey-Kellogg & Ramakrishnan 2001).

1. Given lower-level objects L, *aggregate*, *classify*, and *redescribe* them into higher-level objects H.
2. *Aggregate* higher-level objects H into a neighborhood graph G_H localizing potential correspondence.
3. Apply an *analogy predicate* to relate constituent lower-level objects of neighboring higher-level objects. *analogize* : $(G_H, a) \mapsto \{\{l_1, l_2, w\} \mid \{h_1, h_2\} \in G_H, l_1 \in h_1, l_2 \in h_2, w = a(l_1, l_2) \neq \perp\}$ where $l_i \in h_i$ represents constituency and a returns $\perp$ for no edge or else an edge label.
4. Apply a *correspondence abstraction function* to establish correspondence between higher-level objects based on the analogy on their constituent lower-level objects. *correspond* : $(G_L, G_H, c) \mapsto \{\{h_1, h_2, w\} \mid \{h_1, h_2\} \in G_H, w = c(G_L[\ell(h_1) \cup \ell(h_2)]) \neq \perp\}$ where $\ell(\cdot)$ obtains constituents, $G_L[\cdot]$ is the subgraph for the given nodes, and c returns $\perp$ for no edge or else an edge label.

Table 1: Qualitative correspondence analysis mechanism, including formal definitions of new operators.

Qualitative Analysis of Correspondence

Our correspondence mechanism builds on the relationship between lower-level and higher-level objects in a SAL hierarchy (refer again to Fig. 2). The mechanism has two key steps: (i) establish *analogy* as a relation among lower-level constituents of higher-level objects; (ii) establish *correspondence* between higher-level objects as an *abstraction* of the analogy between their constituents. For example, in object recognition, analogy might match image and template features, and correspondence might abstract the analogy as a rigid-body transformation. Similarly, in level curve analysis, analogy might match sample points on neighboring curves by location and local curvature, and correspondence might abstract the match as a parameterized deformation of spline representations of the curves. The analogy between constituents is well-defined only because of the context of the higher-level objects; higher-level correspondence then captures a more global view of the local matches.

Tab. 1 outlines our correspondence mechanism. Traditional SAL operators collect and abstract groups of lower-level objects into higher-level objects, and establish pairs of higher-level objects for which correspondence is to be considered. Two pieces of domain knowledge are then applied:

Analogy predicate indicates pairs of lower-level objects that are analogous with respect to the higher-level objects they comprise. In Fig. 1, an analogy might seek to connect points on an inner curve with nearby points on an outer curve. Examples of predicates include testing feature values for each pair of lower-level objects, hashing indices in the higher-level objects' local coordinate systems, or explicitly constructing a spatial relation such as a triangulation. The predicate can enforce bijective analogy if appropriate. The *analogize* operator in Tab. 1 applies a function a for the constituent objects $l_1 \in h_1$ and $l_2 \in h_2$ of higher-level object pairs h_1 and h_2; it returns a labeled graph containing analogous object pairs. It takes as input a higher-level graph G_H in order to localize comparisons to only appropriate higher-level object pairs.

Correspondence abstraction function abstracts an analogy relation on lower-level objects into a description of higher-level object correspondence. In Fig. 1, the abstraction might capture the fact that one curve is nicely contained in another, or that two curves merge into a third. As another example, the analogy between constituents of two objects might be abstracted in terms of a rigid-body transformation or parameterized deformation between the objects. In SAL terms, correspondence abstraction simply packages up the details of an analogy on object constituents into a labeled graph on the objects. The *correspond* operator in Tab. 1 performs correspondence abstraction from an analogy G_L, applying a function c to the subgraphs of G_L on the constituent objects $\ell(h_1)$ and $\ell(h_2)$ of higher-level objects h_1 and h_2; it returns a labeled graph on the higher-level objects. As with *analogize*, a higher-level graph G_H localizes abstraction to related higher-level object pairs.

An important consequence of considering an analogy relation holistically (rather than a single related pair at a time) is the ability to compute global properties of the overall correspondence. For example, in Fig. 1, the abstraction allows noting whether a curve is related to a single curve (containment) or multiple curves (merge). This allows the significant event of *discontinuity* to be detected via a break in correspondence. Similarly, the abstraction could test the quality of correspondence, for example computing root-mean squared distance (RMSD) or a Hausdorff metric between locations or features (e.g. local curvature) of related objects.

An aggregation/correspondence hierarchy establishes a distribution of possible high-level models for an input instance, thereby posing a *model selection problem*: choose the one that (e.g. in a maximum-likelihood sense) best matches the data. Our mechanism supports model selection in two key ways. (i) The operators estimate and optimize confidence in correspondence. Since correspondence implies mutual support among parts of a model, it can allow relatively high-confidence model selection even with sparse, noisy data. (ii) The operators bridge the lower-/higher-level gap. This allows weaknesses and inconsistencies detected in higher-level correspondence to focus lower-level data collection to be maximally effective for model disambiguation.

Applications in Scientific Computing

We present two case studies applying our analysis framework to scientific computing domains. For each we describe the underlying numerical analysis problem, our particular solution approach, and results. To the best of our knowledge, these are the *first* systematic algorithms for performing complete imagistic analyses (as opposed to relying on human visual inspection (Chaitin-Chatelin & Frayssé 1996)), and which focus data collection and evaluate models until a high-confidence model is obtained.

Matrix Spectral Portrait Analysis

Our first case study focuses on the previously introduced task of matrix spectral portrait analysis (Fig. 1). Formally, the spectral portrait of a matrix $\mathcal{A}$ is defined as:

$$\mathcal{P}(z) = \log_{10} \|\mathcal{A}\|_2 \; \|(\mathcal{A} - zI)^{-1}\|_2, \tag{1}$$

where I is the identity matrix. The singularities of this map are located at the eigenvalues of the matrix, and the analysis determines the sensitivity of computation to numerical imprecision by analyzing how the map decreases (from ∞) moving away from the eigenvalues. As discussed in the introduction, the region enclosed by a level curve of a certain magnitude contains all points that act as "equivalent" eigenvalues under perturbations of that magnitude. More precisely, the region inside a contour for a given perturbation level k is the set of the eigenvalues of all the perturbed matrices $\mathcal{A} + E$ where E is a matrix with $\|E\|_2 \leq k\|\mathcal{A}\|_2$. The level curves are labeled with the negative logarithm (base 10) of the perturbation level (e.g. the curve for $k = 10^{-7}$ is labeled as -7), so a large label indicates a small perturbation, and appears close to the eigenvalue in the portrait. For example, Fig. 1 illustrates that the eigenvalues at 2 and 3 are the most sensitive to perturbation, but under a large enough perturbation, the eigenvalue at 4 is indistinguishable from them, and under a very large perturbation, even the eigenvalue at 1 is equivalent to the rest. This illustrates that sensitivity analysis is approached by identifying values at which level curves merge.

Tab. 2 describes qualitative correspondence analysis of spectral portraits. Data are collected by computing Eq. 1; the analysis determines perturbation equivalence of eigenvalues by detecting curve merges via level curve correspondence. The first aggregation level generates samples on a coarse regular grid around the known eigenvalue locations, and then interpolates and abstracts level curves as in the spatial aggregation example. The second aggregation level finds correspondence among these curves from a Delaunay triangulation analogy of their constituent points. In abstracting the correspondence, it evaluates what fractions of points on the curves had partners (indicating that the curves were well-matched), as well as how many sample points are between the curves (providing evidence that contours do not merge at a smaller perturbation). (See below for an example.) It tracks correspondence outward from the eigenvalues to establish a model of merge events, using the correspondence abstraction metrics to gain confidence in such a model. Ambiguity-directed sampling generates additional data in order to separate curves and to ensure that each eigenvalue pair is merged at some perturbation.

Input: matrix $\mathcal{A}$, eigenvalues E, perturbation levels V.
Output: $\{(E_i, E_j, v_{ij})\}$ such that eigenvalues E_i and E_j are equivalent with respect to perturbation of $v_{ij} \in V$.

Level one:
- Data collection: Eq. 1.
- Initial samples P: points on coarse regular grid.
- Output: level curves C.
- Aggregation: aggregate grid; interpolate points I at values in V; classify by perturbation; redescribe into curves.
- Aggregation 2: G_I = triangulation of points I.

Level two:
- Input: curves C.
- Output: problem output $\{(E_i, E_j, v_{ij})\}$.
- Aggregation: $(C_k, C_l) \in G_C$ iff constituent points are neighbors in G_I.
- Correspondence:
  - Analogy: cross-curve neighbors in G_I.
  - Abstraction: $C_k, C_l \mapsto (m_k, m_l, \theta_{kl})$, for m_k% and m_l% constituent points matched, and samples $P' \subseteq P$ between C_k and C_l separated by no more than θ_{kl} in angle around the enclosed eigenvalue.
- Model evaluation: follow correspondence outward from each pair of eigenvalues (i, j); evaluate confidence with respect to (m_k, m_l, θ_{kl}).
- Sampling:
  - When no (E_i, E_j, v_{ij}) for some (i, j), expand grid.
  - When some θ_{kl} too large, subsample on finer grid.

Table 2: Correspondence mechanism instantiation for spectral portrait analysis.

Fig. 3 demonstrates the application of the mechanism to the companion matrix of the polynomial $(x-1)^3(x-2)^3(x-3)^3(x-4)$ (Chaitin-Chatelin & Frayssé 1996) (see also Fig. 1). Output models are depicted with "merge trees" qualitatively representing the perturbation levels at which eigenvalues become equivalent (e.g. the bottom merge tree in Fig. 3 indicates that eigenvalues at 2 and 3 are indistinguishable under a perturbation of 10^{-8} or more). *A priori*, a double-factorial number of binary merge trees are possible. The approach presented here eliminates almost all of them without even considering them. Instead, it only considers one plausible tree for a given number of samples, and decides whether or not the merge events captured in the tree

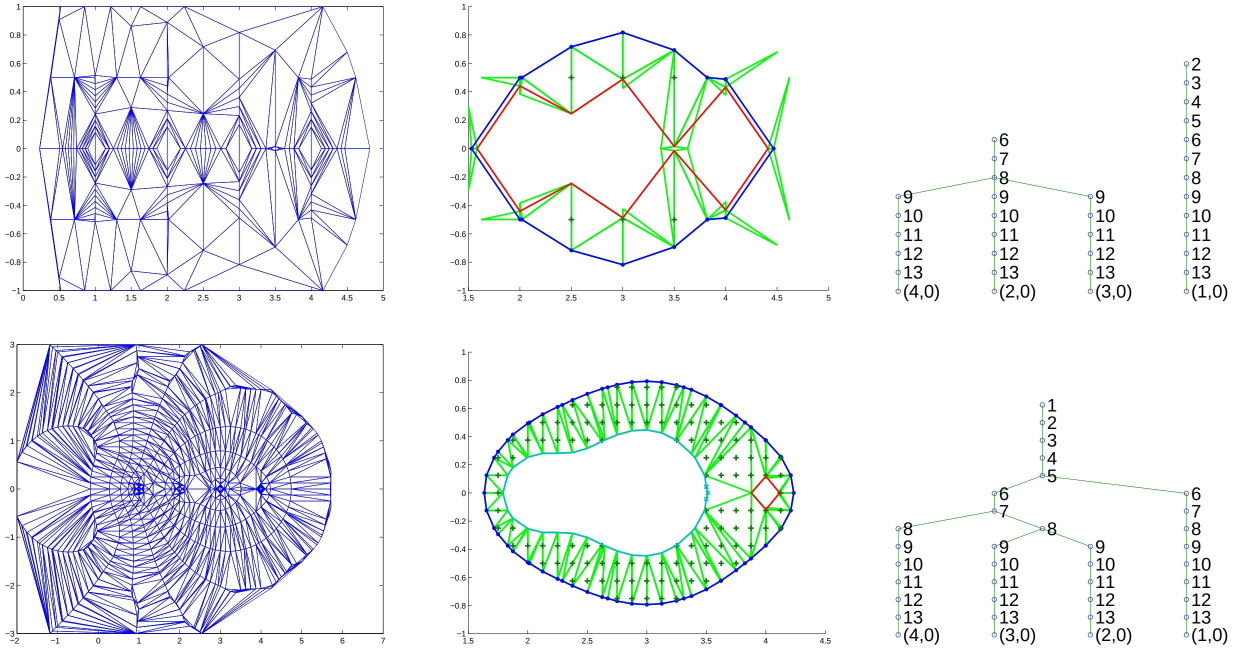

Figure 3: Example of correspondence analysis of spectral portraits for (top) a small, coarse grid, and (bottom) an extended, subsampled grid. (left) Delaunay triangulation analogy for interpolated points comprising contours. (middle) Analogy at an example merge event; separating samples marked with "+". (right) Curve merge tree: eigenvalues at bottom; node for curves labeled with perturbation level. Merge events indicate at what perturbation level descendant eigenvalues are indistinguishable.

are ambiguous. If ambiguity is detected, it either subsamples or expands, as described in Tab. 2. After a small number of runs with successively more samples, it converges to a highly confident tree, essentially declaring that any other model that would be proposed would be highly inconsistent with the data.

The initial grid has at least one sample between each eigenvalue (in the example, a resolution of 0.5) and extends one unit beyond the bounding box of the eigenvalues. Some amount of correspondence is found even with this coarse grid (top of Fig. 3). The model merge tree shown is the best one, but the ambiguity is made clear by the model evaluation: only a few samples separate the curves (i.e. P' is small, so the curves might have merged earlier) and no curve surrounds all eigenvalues. A distribution of confidence would thus be relatively flat over multiple possible models; e.g. all models merging the currently-disconnected eigenvalue at a large perturbation are equally good. As a result, ambiguity-directed sampling computes additional points on a finer, larger grid (bottom of Fig. 3), yielding high confidence in the (correct) curve merge tree shown, since many samples provide evidence for the merge events.

We have applied this approach to a variety of polynomial companion matrices with different numbers and spacings of roots; in each case, the correspondence mechanism correctly identifies the correct model with high confidence after 1–3 subsamples and 1–3 grid expansions.

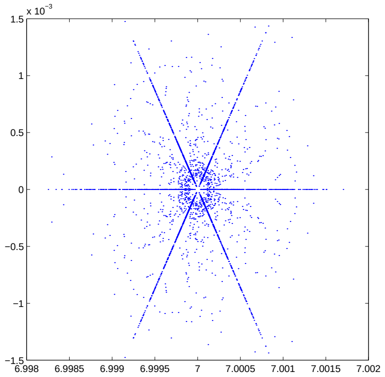

Figure 4: Superimposed spectra for assessing the Jordan form of the Brunet matrix.

Qualitative Computation of Jordan Forms

Our second case study focuses on analysis of the *Jordan decomposition* of a matrix. Matrix decomposition is an important technique, revealing pertinent features of a matrix and supporting algorithmic techniques in areas including data analysis, PDEs, and linear algebra. The Jordan decomposition reveals the eigenstructure of a matrix as follows. Consider a matrix $\mathcal{A}$ of dimension n that has $r \leq n$ independent eigenvectors with eigenvalues λ_i of multiplicity ρ_i. The Jordan decomposition of $\mathcal{A}$ contains r upper triangular "blocks," as revealed by the diagonalization:

$$\mathcal{B}^{-1}\mathcal{A}\mathcal{B} = \begin{bmatrix} \mathcal{J}_1 & & & \\ & \mathcal{J}_2 & & \\ & & \cdot & \\ & & & \mathcal{J}_r \end{bmatrix}, \ \mathcal{J}_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & 1 & \\ & & \cdot & 1 \\ & & & \lambda_i \end{bmatrix}$$

where $\mathcal{B}$ is the diagonalizing matrix. The typical approach to computing the Jordan form leads to a numerically unstable algorithm (Golub & Van Loan 1996); taking extra care usually requires more work than the original computation! Recently, however, a data-driven approach, inferring multiplicity from a geometric analysis of eigenvalue perturbations, has proved successful (Chaitin-Chatelin & Frayssé 1996). It is well known that the *computed eigenvalues* corresponding to the actual value λ_i are given by:

$$\lambda_i + |\delta|^{\frac{1}{\rho_i}} e^{\frac{i\phi}{\rho_i}}, \qquad (2)$$

where λ_i is of multiplicity ρ_i, and the phase ϕ of the perturbation δ ranges over $\{2\pi, 4\pi, \ldots, 2\rho_i\pi\}$ if δ is positive and over $\{3\pi, 5\pi, \ldots, 2(\rho_i + 1)\pi\}$ if δ is negative. The insight of (Chaitin-Chatelin & Frayssé 1996) is to graphically *superimpose* numerous such perturbed calculations so that the aggregate picture reveals eigenvalue multiplicity. The phase variations imply that computed eigenvalues lie on the vertices of a regular polygon with $2\rho_i$ sides, centered on λ_i, and with diameter influenced by $|\delta|$. For example, Fig. 4 shows perturbations for the 8-by-8 Brunet matrix with Jordan structure $(-1)^1(-2)^1(7)^3(7)^3$ (Chaitin-Chatelin & Frayssé 1996), for $\delta \in [2^{-50}, 2^{-40}]$. The six "sticks" around the eigenvalue at 7 clearly reveal that its Jordan block is of size 3.[2] The "noise" in Fig. 4 is a consequence of having two Jordan blocks with the same eigenvalue and size, and a "ring" phenomenon studied in (Edelman & Ma 1998); we do not attempt to capture these effects in this paper.

Tab. 3 describes qualitative correspondence analysis of Jordan form. Data are collected by randomly perturbing at a specified magnitude δ; the analysis determines multiplicity by detecting *symmetry* correspondence in the samples. The first aggregation level collects the samples for a given δ into triangles. The second aggregation level finds congruent triangles via geometric hashing (Lamdan & Wolfson 1988), and uses congruence to establish analogy among triangle vertices. Correspondence abstracts the analogy into a rotation about a point (the eigenvalue), and evaluates whether each point rotates onto another and whether matches define regular polygons as required by the underlying math above. Ambiguity-directed sampling collects additional random samples as necessary. A third level then compares rotations across different perturbations, re-visiting perturbations or choosing new perturbations in order to disambiguate. The output is a symbolic description of the Jordan form: the location of the eigenvalue and its multiplicity.

Fig. 5 demonstrates this mechanism on the Brunet matrix discussed above. The top part uses a small set of sample points, while the bottom two parts use a larger set and illustrate a good vs. bad correspondence. With a small number of samples, multiple models are consistent with the data, as indicated by the model evaluation metric. With more samples, the degrees of freedom are rapidly pinned down and the confidence distribution over models becomes peaked at the correct one. However, as the number of samples increases, so does the risk of model "hallucination" — finding some subset of points that by chance happen to correspond, as in bottom of Fig. 5. This illustrates the importance of monitoring relative model confidence and controlling the sampling to avoid over-sampling.

To study the effect of sampling strategy, we organized data collection into rounds of 6–8 samples each and experimented with three policies on where next to collect data after completing a round: (1) at the same perturbation level, (2) at a higher perturbation level, or (3) at the same perturbation level unless the number of posited models increased (thereby avoiding hallucination). We tested 10 matrices across 4–10 perturbation levels each, as described in (Chaitin-Chatelin & Frayssé 1996). We 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. Policy 1 required an average of 1 round at a tolerance of 0.1, up to 2.7 rounds at 0.5. Even with a large number of models proposed, additional data quickly weeded out bad models. Policy 2 fared better only for cases where policy 1 was focused on lower perturbation levels, and policy 3 was preferable only for the Brunet-type matrices. In other words, there is no real advantage to moving across perturbation levels! In retrospect, this is not surprising since our Jordan form computation treats multiple perturbations (irresp. of level) as independent estimates of eigenstructure.

Discussion

Our general mechanism for uncovering and utilizing correspondence has proved successful in active data mining for challenging problems in scientific computing. The mechanism leverages properties such as locality, continuity, and decomposability, which are exhibited by the applications studied here as well as many physical systems. Decomposability and locality allow us to qualify correspondence in a manner that drives data collection. Continuity allows correspondence to obtain confidence in a model even with sparse, noisy data: consistent matches among nearby constituents mutually reinforce each other, allowing correspondence abstraction to detect and filter out inconsistent interpretations.

Correspondence can be an appropriate analysis tool for a variety of reasons. In the spectral portrait application, level curves summarize groups of eigenvalue computations, so their high-level correspondence aids characterization of the underlying computation. In the Jordan form application, however, the higher-level entity has no significance as a geometric object in numerical analysis terms, but correspondence is applicable due to the semantics of superposition. This semantics also leads to phenomena such as hallucination (given enough samples, any pattern can be found), requiring a more careful treatment of decomposition.

Our work is similar in spirit to that of (Huang & Zhao 1999) for weather data interpretation, and can be seen as a

[2] The multiplicity of the second eigenvalue at 7 is revealed at a smaller perturbation level.

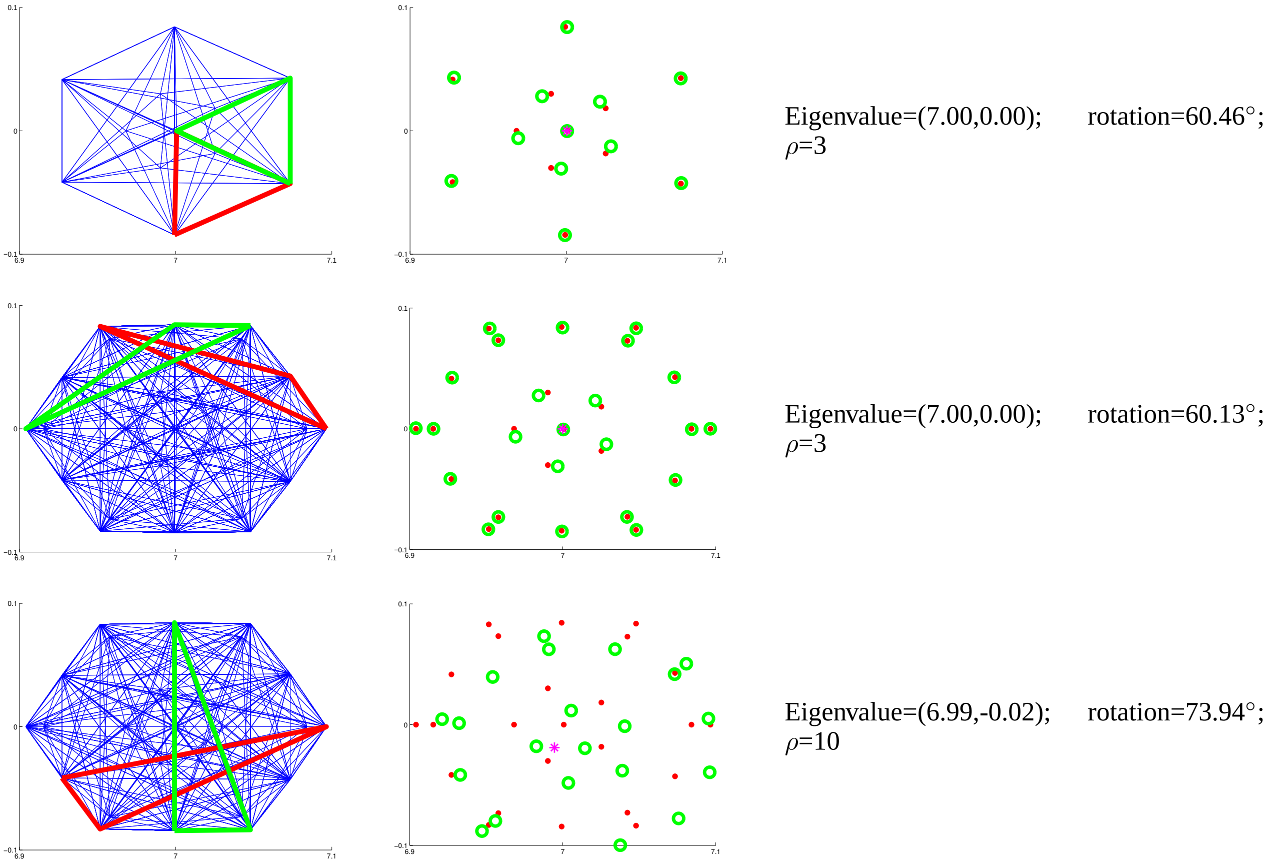

Figure 5: Example of correspondence analysis of Jordan form, for (top) small sample set; (middle) larger sample set; (bottom) larger sample set but lower-scoring model. (left) Approximately-congruent triangles. (middle) Evaluation of correspondence in terms of match between original (red dots) and rotated (green circles) samples. (right) Associated model of Jordan form.

significant generalization, formalization, and application of techniques studied there for finding correspondence in meteorological data. Similarly, our correspondence framework captures and generalizes the computation required in object recognition, allowing the body of research developed there to be applied to a broader class of applications, such as experimental algorithmics. Compared to traditional manual analyses of graphical representations like spectral portraits, the algorithmic nature of our approach yields advantages such as model evaluation and targeted sampling. As with compositional modeling (Falkenhainer & Forbus 1991), we advocate targeted use of domain knowledge, and as with qualitative/quantitative model selection (e.g. (Capelo, Ironi, & Tentoni 1998)), we seek to determine high level models for empirical data. Our focus is on problems requiring particular forms of domain knowledge to overcome sparsity and noise in spatial datasets. A possible direction of future work is to explore if the inclusion-exclusion methodology popular in grid algorithms (Bekas *et al.* 2001) is also useful for tracking correspondence.

Our long-term goal is to study data collection policies and their relationships to qualitative model determination. The notion of estimating problem-solving performance by collecting data (and vice versa) is reminiscent of reinforcement learning (Boyan & Moore 2000) and active learning (Cohn, Ghahramani, & Jordan 1996). The decomposable nature of SAL computations promises to (i) support the design of efficient, hierarchical algorithms for model estimation and (ii) provide a deeper understanding of the recurring roles that correspondence plays in spatial data analysis.

Acknowledgments

We would like to thank the reviewers for very helpful comments. This work is supported in part by US National Science Foundation grants to CBK (IIS-0237654) and NR (EIA-9974956, EIA-9984317, and EIA-0103660).

References

Bailey-Kellogg, C., and Ramakrishnan, N. 2001. Ambiguity-Directed Sampling for Qualitative Analysis of Sparse Data from Spatially Distributed Physical Systems. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, 43–50.

Bailey-Kellogg, C., and Zhao, F. 1999. Influence-Based Model Decomposition. In *Proceedings of the Sixteenth*

National Conference on Artificial Intelligence (AAAI'99), 402–409.

Bailey-Kellogg, C., and Zhao, F. 2001. Influence-Based Model Decomposition for Reasoning about Spatially Distributed Physical Systems. *Artificial Intelligence* Vol. 130(2):pages 125–166.

Bailey-Kellogg, C.; Zhao, F.; and Yip, K. 1996. Spatial Aggregation: Language and Applications. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, 517–522.

Bekas, C.; Kokiopulou, E.; Koutis, I.; and Gallopoulos, E. 2001. Towards the Effective Parallel Computation of Matrix Pseudospectra. In *Proceedings of the International Conference on Supercomputing (ICS)*, 260–269.

Boyan, J., and Moore, A. 2000. Learning Evaluation Functions to Improve Optimization by Local Search. *Journal of Machine Learning Research* Vol. 1:pages 77–112.

Capelo, A.; Ironi, L.; and Tentoni, S. 1998. Automated Mathematical Modeling from Experimental Data: An Application to Material Science. *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 28(3):pages 356–370.

Chaitin-Chatelin, F., and Frayssé, V. 1996. *Lectures on Finite Precision Computations*. SIAM Monographs.

Cohn, D.; Ghahramani, Z.; and Jordan, M. 1996. Active Learning with Statistical Models. *Journal of Artificial Intelligence Research* Vol. 4:pages 129–145.

Edelman, A., and Ma, Y. 1998. Non-Generic Eigenvalue Perturbations of Jordan Blocks. *Linear Algebra & Applications* Vol. 273(1-3):pages 45–63.

Falkenhainer, B., and Forbus, K. 1991. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence* Vol. 51(1-3):pages 95–143.

Golub, G., and Van Loan, C. 1996. *Matrix Computations*. Johns Hopkins University Press. Third Edition.

Huang, X., and Zhao, F. 1999. Relation-Based Aggregation: Finding Objects in Large Spatial Datasets. In *Proceedings of the 3rd International Symposium on Intelligent Data Analysis*.

Lamdan, Y., and Wolfson, H. 1988. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. In *Proceedings of ICCV*, 238–249.

Ordóñez, I., and Zhao, F. 2000. STA: Spatio-Temporal Aggregation with Applications to Analysis of Diffusion-Reaction Phenomena. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI'00)*, 517–523.

Ramakrishnan, N., and Ribbens, C. 2000. Mining and Visualizing Recommendation Spaces for Elliptic PDEs with Continuous Attributes. *ACM Transactions on Mathematical Software* Vol. 26(2):pages 254–273.

Yip, K., and Zhao, F. 1996. Spatial Aggregation: Theory and Applications. *Journal of Artificial Intelligence Research* Vol. 5:pages 1–26.

Input: matrix $\mathcal{A}$, perturbations $\{\delta_1, \dots, \delta_m\}$, region R.
Output: eigenvalue λ and multiplicity ρ in region R.

Level one:

- Data collection: for δ_i, compute random normwise perturbation of $\mathcal{A}$ as $a_{ij} \pm 2^{(1-\delta_i)} \|\mathcal{A}\|_\infty$, yielding P_i.
- Initial samples: At some level i.
- Output: triangles T_i.
- Aggregation: triangulate P_i (for efficiency, require 2 vertices on convex hull).

Level two:

- Input: Triangles T_i.
- Output: set of rotations (x, y, θ).
- Aggregation: congruent triangles by geom. hashing.
- Correspondence:
  - Analogy: for each pair of congruent triangles t_j, t_k compute superimposing rotation $R = (x, y, \theta)$; apply to all points and find closest match $a(\cdot)$ for each.
  - Abstraction: $(t_j, t_k) \mapsto (R, d, r)$, for specific R and associated a, with quality metrics
    - d: distance between points and transformed analogs: $\sum_{p \in P_i} \|p - R(a(p))\|$
    - r: regularity of polygon, by comparing distance between point and its partners in both directions: $\sum_{p \in P_I} (\|p - a(p)\| - \|p - a^{-1}(p)\|$
- Model evaluation: confidence with respect to d and r; priors support rotations around (x, y) in convex hull of P_i and by θ corresponding to "reasonable" multiplicity.
- Sampling: for multiple "good" models, collect additional random samples at perturbation δ_i.

Level three:

- Input: $\{(x_{ij}, y_{ij}, \theta_{ij})\}$ over models (j) from chosen perturbation levels (i).
- Output: (λ, ρ).
- Aggregation: clustering in (x, y, θ)-space.
- Model evaluation: for $\lambda = (x, y), \rho = \pi/\theta$, take joint probability over i, j of $(x_{ij}, y_{ij}, \theta_{ij}) \approx (x, y, \theta)$.
- Sampling: for high entropy in model evaluation, add samples and re-evaluate at outlier δ_j or try new δ_k.

Table 3: Correspondence mechanism instantiation for Jordan form analysis.