Gaussian Processes for Active Data Mining of Spatial Aggregates

Naren Ramakrishnan[†], Chris Bailey-Kellogg[#], Satish Tadepalli[†], and Varun N. Pandey[†]

[†] Department of Computer Science
Virginia Tech, Blacksburg, VA 24061
{naren,stadepal,vnpandey}@cs.vt.edu

[#] Department of Computer Science
Dartmouth College, Hanover, NH 03755
cbk@cs.dartmouth.edu

Abstract

We present an active data mining mechanism for qualitative analysis of spatial datasets, integrating identification and analysis of structures in spatial data with targeted collection of additional samples. The mechanism is designed around the spatial aggregation language (SAL) for qualitative spatial reasoning, and seeks to uncover high-level spatial structures from only a sparse set of samples. This approach is important for applications in domains such as aircraft design, wireless system simulation, fluid dynamics, and sensor networks. The mechanism employs Gaussian processes, a formal mathematical model for reasoning about spatial data, in order to build surrogate models from sparse data, reason about the uncertainty of estimation at unsampled points, and formulate objective criteria for closing-the-loop between data collection and data analysis. It optimizes sample selection using entropy-based functionals defined over spatial aggregates instead of the traditional approach of sampling to minimize estimated variance. We apply this mechanism on a global optimization benchmark comprising a testbank of 2D functions, as well as on data from wireless system simulations. The results reveal that the proposed sampling strategy makes more judicious use of data points by selecting locations that clarify high-level structures in data, rather than choosing points that merely improve quality of function approximation.

Introduction

Many scientific and engineering applications require analysis of spatial datasets derived from computer simulations or field data, e.g., wireless system simulations, aircraft design configuration spaces, fluid dynamics simulations, and sensor network optimization. While a wealth of data is potentially available in these domains, there is significant cost and time involved in conducting simulations or setting up experimental apparatus. Thus in practice, these applications require qualitative spatial analysis and reasoning, providing high-level interpretations of a sparse set of data. Based on an initial, possibly imperfect, analysis, additional data can be collected to clarify ambiguities. Since the computational scientist has control over *where* data can be collected, it is prudent to focus data collection in only those regions that are deemed important to support a high-level analysis objective. Multiple rounds of analysis and sampling can be conducted in this fashion.

As a concrete example, consider the characterization of WCDMA (wideband code-division multiple access) wireless system configurations for a given indoor environment. A configuration comprises many adjustable parameters, and the goal of wireless system characterization is to assess the relationship between these parameters and performance metrics such as BER (bit error rate), a measure of the number of bits transmitted in error using the system. When a wireless engineer designs a system for a given indoor environment, he or she sets an acceptable performance criterion for BER (e.g., 10^{-3} for a system designed to carry voice traffic, stricter thresholds for data traffic) and seeks a region in the configuration space that can satisfy this criterion (see Fig. 1). To collect the data necessary for analyzing configuration spaces, the engineer either performs a costly Monte Carlo simulation (where a model of radio propagation in the wireless channel is embedded inside a system-wide model encapsulating wireless protocols and communications standards), or installs channel sounding equipment and system instrumentation in the environment, and actually enacts usage scenarios. In either approach, it is not feasible to first organize a voluminous body of data and subsequently analyze the collected dataset. It is thus imperative that we interleave data collection and data analysis and focus sampling at only those locations that maximize well-defined notions of relevance and utility. Importantly, we will not need to sample the entire configuration space, only enough so as to identify and characterize a region with acceptable confidence.

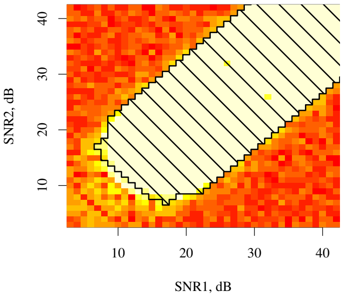

Figure 1: Analyzing configuration spaces from wireless system simulations. The shaded region denotes the largest portion of the configuration space where we can claim, with confidence at least 99%, that the average bit error rate (BER) is acceptable for voice-based system usage.

In this paper, we develop an active mining mechanism based on the spatial aggregation language (SAL; [Bailey-Kellogg et al., 1996]), a generic framework for qualitative spatial reasoning, and Gaussian pro-

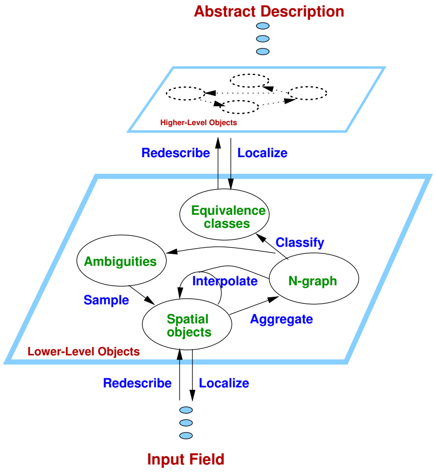

Figure 2: SAL uncovers multi-level spatial aggregates by employing a small set of operators (a spatial reasoning "vocabulary") utilizing suitable domain knowledge. Basic bottom-up flow: computations on spatial objects within an abstraction level are localized within neighborhoods; equivalence classes are formed from neighboring, similar objects; classes of equivalent objects are redescribed into higher-level objects for processing at the next level of abstraction.

cesses (GPs; [Williams, 1998]), a powerful unifying theory for approximating and reasoning about datasets. SAL uncovers successive multi-level aggregates of spatial data, and Gaussian processes provide the 'glue' that enables us to perform active mining on these aggregates. In particular, they aid in (i) creation of *surrogate models* from data using a sparse set of samples (for cheap generation of dense approximate datasets), (ii) reasoning about the uncertainty of estimation at unsampled points, and (iii) formulation of objective criteria for active data collection. This approach enables us to design sampling strategies that bridge higher-level quality metrics of spatial structures (e.g., entropy) with lower-level considerations of data samples (e.g., locations and fidelity).

Spatial Aggregation Language

The Spatial Aggregation Language (SAL) [Bailey-Kellogg et al., 1996] supports structure discovery in spatial datasets 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 Fig. 2) to form higher-level abstractions and mine patterns. The *primitives* in SAL are contiguous regions of space called *spatial objects*; the *compounds* are (possibly structured) collections of spatial objects; the *abstraction mechanisms* connect collections at one level of abstraction with single objects at a higher level. This vocabulary has proved effective for expressing the mechanisms required to uncover multi-level structures in spatial datasets in applications ranging from diffusion-reaction morphogenesis [Ordóñez and Zhao, 2000] to decentralized control design [Bailey-Kellogg and Zhao, 2001], to matrix perturbation analysis [Bailey-Kellogg and Ramakrishnan, 2004].

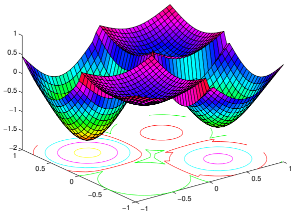

Figure 3: de Boor's 'pocket' function in 2D, depicting contours around basins of local minima.

SAL Pocket Finder

Let us consider a specific qualitative spatial reasoning task motivated by the wireless study — determining the number and locations of *pockets*, or basins of local minima, in a vector field. Fig. 3 illustrates four pockets in a field defined by Carl de Boor's function in 2D (from [Ramakrishnan and Bailey-Kellogg, 2002]). This function is a well-known benchmark for global optimization (esp. in high dimensions), but we focus here on a somewhat different objective of characterizing the high-level structure of the field. The algorithmic encoding of the calculus definition of local minima suggests that the four pockets in Fig. 3 can be identified via convergent flows in the gradient underlying the vector field. Let us assume we are given a dense set of samples covering the region of interest. Fig. 4 illustrates an example of key spatial aggregation operations:

(a) Establish the input field, here by calculating the gradient field (normalized, since we're interested only in direction in order to detect convergence).

(b) Localize computation with a *neighborhood graph*, so that only spatially proximate objects are compared. Here, an 8-adjacency neighborhood graph is employed, which results in somewhat 'blocky' streamlines but fast computation.

(c)–(f) Use a series of local computations to find *equivalence classes* of neighboring objects with similar features. Here, we systematically eliminate all neighborhood graph edges but those whose directions best match the vector direction at both endpoints. 'Forward neighbor' computation compares graph edge direction with the average of the vector directions, and keeps only those that are similar enough (implemented as a cosine angle similarity threshold). 'Best forward neighbor' at junction points then selects from among these neighbors, by a third metric combining similarity in direction with closeness in point location. Backward calculations are analogous, but deal with the predecessor along a streamline rather than the successor.

(g) Move up a level in the spatial object hierarchy by *redescribing* equivalence classes into more abstract objects. Here, connected vectors are abstracted into curve objects, which have both a reduced representation and additional semantic properties (e.g. curvature is well-defined).

(h) Apply the same mechanism — aggregate, classify, and redescribe — at the new level, *using the exact same*

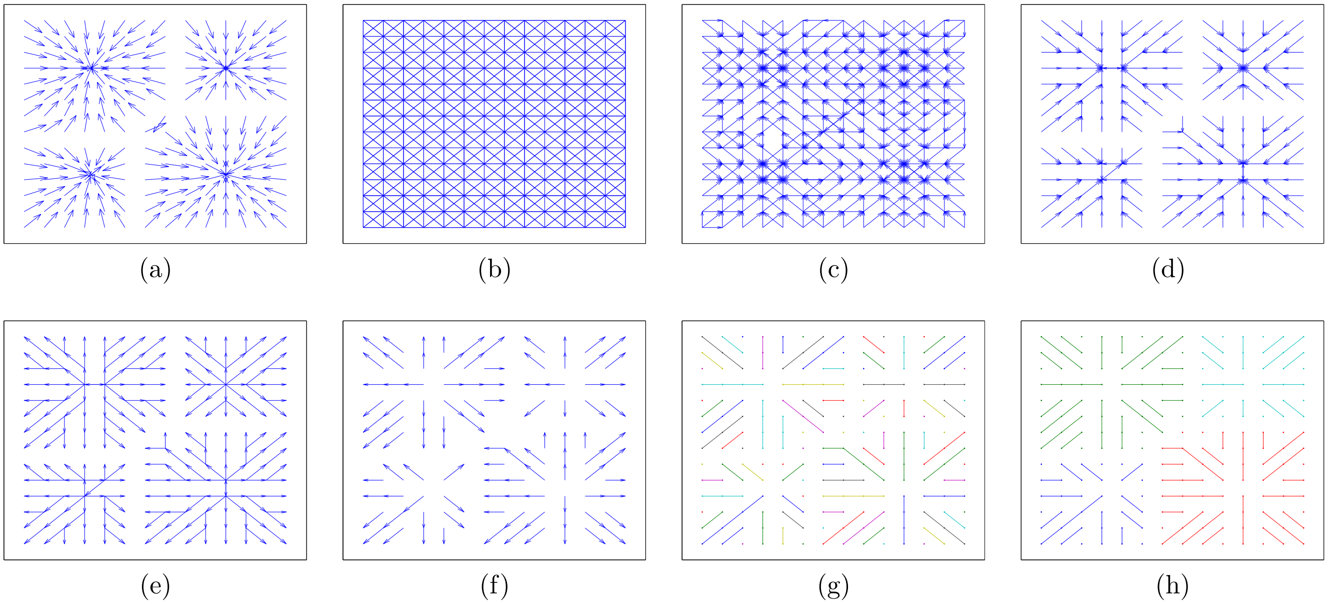
(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4: Example steps in SAL pocket finder based on vector field analysis of de Boor's function. (a) Input vector field. (b) 8-adjacency neighborhood graph. (c) Forward neighbors. (d) Best forward neighbors. (e) Ngraph transposed from best forward neighbors. (f) Best backward neighbors. (g) Resulting adjacencies redescribed as curves. (h) Higher-level aggregation and classification of curves whose flows converge. When the vector field is the gradient of a pocket function, these are the pockets.

operators but with different metrics. Here, curves are grouped into coherent pockets with convergent flow. Neighborhood (not shown) is derived from neighborhood of constituent vectors, and equivalence tests direction of flow for convergence.

Localized computations are integral to SAL, as aggregates are identified by grouping "close-enough, similar-enough" spatial objects. When data is scarce, we can achieve locality by approximating values on a denser field than that provided as input. In particular, we can construct a *surrogate* model — a cheap-to-compute substitute for a complex function — that serves as an approximation to the underlying field with the given samples. We can then use this approximation to generate a dense field of data (e.g., on a uniform grid). One way to build surrogate models relies on Gaussian processes.

Gaussian Processes

Gaussian processes (GPs) are a modeling mechanism with origins in spatial statistics, particularly kriging [Journel and Huijbregts, 1992]. In contrast to global approximation techniques such as least-squares fitting, GPs are local approximation techniques, akin to nearest-neighbor procedures. In contrast to function approximation techniques that place a prior on the form of the function, GP modeling techniques place a prior on *the covariance structures* underlying the data.

The basic idea in GPs is to model a given dataset as a realization of a stochastic process. Formally, a GP is a set of random variables any finite subset of which have a (multivariate) normal distribution. For our purposes, we can think of these variables as spatially distributed (scalar) response variables t_i, one for each 2D location $\mathbf{x_i} = [x_{i1}, x_{i2}]$ where we have collected a data sample. In our vector field analysis application, t_i denotes the modeled response, i.e., the value of de Boor's function at $\mathbf{x_i}$. Given a dataset $\mathcal{D} = \{\mathbf{x_i}, t_i\}, i = 1 \ldots n$, and a new data point $\mathbf{x_{n+1}}$, a GP can be used to model the posterior $P(t_{n+1}|\mathcal{D}, x_{n+1})$ (which would also be a Gaussian). This is essentially what many Bayesian modeling techniques do (e.g., least squares approximation with normally distributed noise) but it is the specifics of how the posterior is modeled that make GPs distinct as a class of modeling techniques.

To make a prediction of t_{n+1} at a point $\mathbf{x_{n+1}}$, GPs place greater reliance on t_i's from nearby points. This reliance is specified in the form of a covariance prior for the process and will be central to how we embed SAL in a broader GP framework:

$$\mathrm{Cov}(t_i, t_j) = \alpha \exp\left(-\frac{1}{2}\sum_{k=1}^{2} a_k (x_{ik} - x_{jk})^2\right) \qquad (1)$$

Intuitively, this function captures the notion that response variables at nearby points must have high correlation. The reader will note that this idea of influence decaying with distance has an immediate parallel to how SAL programs localize computations. In Eq. 1, α is an overall scaling term whereas a_1, a_2 define the length scales for the two dimensions. However, this prior (or even its posterior) does not directly allow us to determine t_j from t_i, since the structure only captures the covariance; predictions of a response variable for new sample locations are thus conditionally dependent on the measured response variables and *their* sample locations. Hence, we must first estimate the covariance parameters (a_1, a_2, and α) from $\mathcal{D}$ and then use these parameters *along with* $\mathcal{D}$ to predict t_{n+1} at $\mathbf{x_{n+1}}$. Since the joint distribution of the response variables $P(t_1, t_2, \cdots, t_{n+1})$ is modeled Gaussian (a mean of zero is often assumed), we can develop the posterior of any desired response variable in terms of its covariances with other points. Specifically, for the covariance formulation above, an estimate of t_{n+1} can be given by:

$$\hat{t}_{n+1} = \mathbf{k}^T \mathrm{Cov}_n^{-1} [t_1, t_2, \cdots, t_n]^T \quad (2)$$

where our uncertainty in this estimate would be:

$$\sigma^2_{\hat{t}_{n+1}} = k - \mathbf{k}^T \mathrm{Cov}_n^{-1} \mathbf{k} \quad (3)$$

Here, $\mathbf{k}^T$ represents the n-vector of covariances with the new data point:

$$\mathbf{k}^T = [\mathrm{Cov}(\mathbf{x_1}, \mathbf{x_{n+1}}) \ , \cdots , \ \mathrm{Cov}(\mathbf{x_n}, \mathbf{x_{n+1}})]$$

and k is the $(n+1, n+1)$ entry of Cov_{n+1}. Eqs. 2 and 3, together, give us both an approximation at any given point and an uncertainty in this approximation; they will serve as the basic building blocks for closing-the-loop between data modeling and higher level analysis functionality.

Learning a GP

Learning the GP parameters $\theta = (a_1, a_2, \alpha)$ can be undertaken in the *maximum likelihood* (ML) and *maximum a posteriori* (MAP) frameworks, or in the true Bayesian setting where we obtain a distribution over values. The log-likelihood for the parameters is given by:

$$\begin{aligned} \mathcal{L} &= \log P(t_1, t_2, \cdots, t_n | \mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n}, \theta) \\ &= c + \log P(\theta) - \frac{n}{2}\log(2\pi) - \frac{1}{2}\log \mid \mathrm{Cov}_n \mid \\ & \quad -\frac{1}{2}[t_1, t_2, \cdots, t_n] \ \mathrm{Cov}_n^{-1} \ [t_1, t_2, \cdots, t_n]^T \quad (4) \end{aligned}$$

To optimize for the parameters, we can compute partial derivatives of the log-likelihood, for use with a conjugate gradient algorithm that operates in the three-space of θ parameters. For larger numbers of parameters, we can resort to the use of Markov Chain Monte Carlo (MCMC) methods [Neal, 1997].

Active Data Mining Strategies

The preceding section showed two important uses of GPs for spatial reasoning: designing a surrogate function for generating a dense surrogate field (via Eq. 2), and assessing uncertainties in our estimates of the function at unsampled points (using Eq. 3). We are now ready to formulate objective criteria for active data selection, a pre-cursor to active mining.

Variance Reducing Designs

A simple strategy for sampling is to target locations to reduce our uncertainty in modeling, i.e., select the location that minimizes the posterior generalized variance of the function. This approach can be seen as optimizing sample selection for the functional:

$$\Phi_V = \frac{1}{2} \log \ \left[\frac{\partial t}{\partial \theta}\right] \ \mathcal{H}^{-1} \ \left[\frac{\partial t}{\partial \theta}\right]^T \quad (5)$$

where $\left[\frac{\partial t}{\partial \theta}\right]$ is the (row) vector of sensitivities w.r.t. each GP parameter computed at a sample location, and $\mathcal{H}$ is the Hessian (second order partial derivatives) of t, again w.r.t. the parameters. A straightforward derivation will show that optimizing Φ_V suggests a location whose 'error bars' σ^2 are highest.

To implement this strategy, we can adopt either a block design (optimize for K locations simultaneously), or apply it sequentially to determine one extra sampling location at a time. Fig. 5 shows the use of the sequential sampling strategy for the pocket function of Fig. 3 and concomitant results from pocket analysis of the surrogate model data. At each step, we determine the best sample location (from among unsampled locations on a regular grid of 21 × 21), build the GP model from the data collected thus far, and apply our SAL-based vector aggregation mechanism to the gradient field derived from the function values.

The initial design has one point in the center of each quadrant, and one at the center. Not surprisingly, we find a significant number (16) of basins in the gradient field. The next four points added are actually at the corners; this is because estimated variances are typically high toward the boundaries of an interpolation region. As MacKay points out [MacKay, 1992], such a metric has a tendency to 'repeatedly gather data at the edges of the input space.' The emphasis on overall quality of function approximation more than data analysis is evident from the fact that it takes over 30 points before the SAL pocket finder can infer that there are four pockets. In further experiments not reported here, we have found that pushing the initial points outward (or inward) does not have any appreciable effect on future samplings, and the variance-based metric favors the outer envelope of the design space.

Entropy-Based Functionals

To develop a better active mining strategy, notice that our goal is the identification of regions defined by convergent flows. If we view the SAL program as an information processor that maps a data field into a class field (defined over the same underlying space), then the utility of sampling in a region is directly related to our inferential capabilities about the corresponding region in the class field. Intuitively, we should be more interested in samples that tell us something about the boundary between regions than those that capture the insides of a region, *even though the latter might have high variance in its current estimate.* Repeatedly sampling function values inside an already classified and abstracted region is not as useful as sampling to clarify an emerging boundary classification. This means that we must bridge high-level information about pockets from SAL into a preference of where to collect data.

An idea that suggests itself is to adopt variance-based design, but instead of minimizing the entropy of the data distribution, minimize the entropy of the class distribution as revealed by the SAL pocket finder. By positing a class distribution at each point, based on the class labels occupied by neighboring points, we achieve our

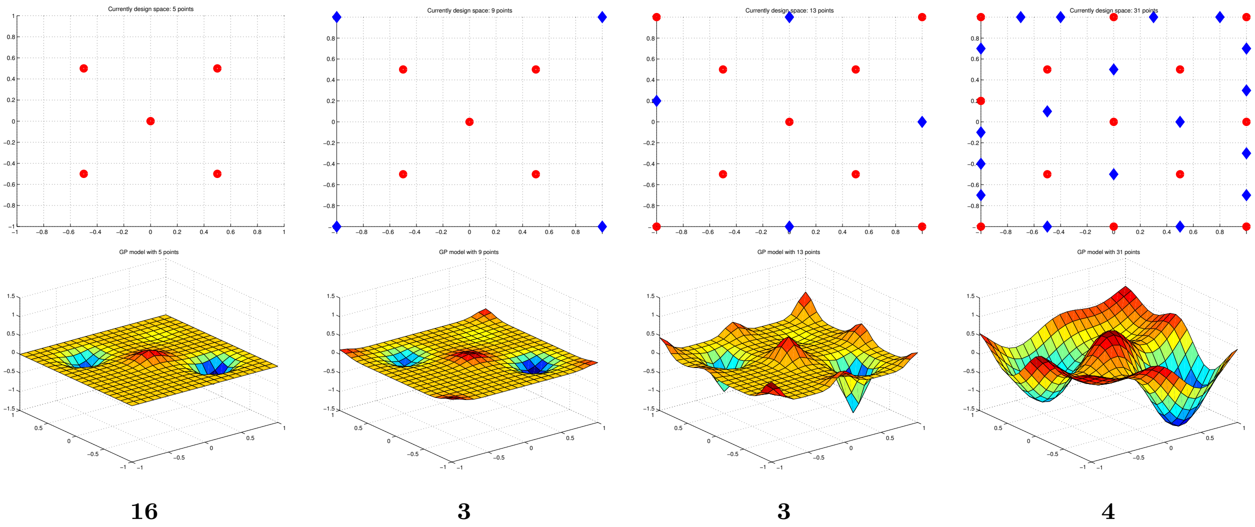

Figure 5: Variance-based sampling. (top row) initial design of 5 points, followed by snapshots taken at later stages (9, 13, and 31 points). Old sample locations are shown with red circles and new locations are shown with blue diamonds. (middle row) GP model fits to the given samples. (bottom row) Number of pockets identified by SAL pocket finder.

goal of ranking locations along region boundaries higher. While this basic strategy appears reasonable, it will repeatedly gather information at the region boundaries, just as variance-based design repeatedly focuses on the edges. So a point with high entropy is a good location to sample only as long as the variance surrounding it is sufficiently high. As our confidence in the data value increases, our preference for this location should decrease even if the class entropy remains large (as it will, if it lies on a boundary). This suggests using class entropy to define a distribution $P_E(\mathbf{x})$ over points, and using that distribution to scale the variance-based design criterion:

$$\Phi_E = \frac{1}{2} \sum_{\mathbf{x}} P_E(\mathbf{x}) \ \log \left[\frac{\partial t}{\partial \theta}\right] \ \mathcal{H}^{-1} \ \left[\frac{\partial t}{\partial \theta}\right]^T \qquad (6)$$

The expression inside the summation contains the same terms as in Eq. 5 but is now evaluated across the design space and scaled by the amount of interest in location $\mathbf{x}$:

$$P_E(\mathbf{x_i}) \propto \sum_{\mathbf{x} \in \mathcal{N}(\mathbf{x_i})} P(C(\mathbf{x})) \ \log P(C(\mathbf{x})) \qquad (7)$$

where $\mathcal{N}(\mathbf{x_i})$ is a neighborhood around $\mathbf{x_i}$, $C(\mathbf{x})$ denotes the (flow) class of point $\mathbf{x}$ as inferred by the SAL algorithm, and $P(C(\mathbf{x}))$ denotes the probability of encountering this class in the neighborhood. The proportionality constant in Eq. 7 must be set to ensure that $\sum P_E = 1$. Formal characterization of this criterion (i.e., its convergence properties) is difficult since $P_E(\mathbf{x})$ changes during every iteration of data analysis, and we do not have a model of how $P_E(\mathbf{x})$ varies across samplings. Operationally, to apply this criterion, we can identify the location that gives the highest information *gain*, given that we are intending to make a measurement at that location. Fig. 6 shows a design that optimizes Φ_E and successfully reveals all four pockets with only 11 points.

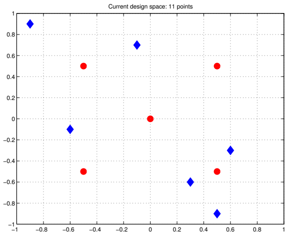

Figure 6: Entropy-based sampling. This strategy picks six additional points, in various quadrants so that the SAL pocket finder identifies four pockets (not shown) when a GP model is constructed using the given points.

Computational Considerations

In addition to data collection costs, the other primary costs to implementing the active mining mechanisms involve the nested optimizations and the necessary matrix computations. There are two optimizations per round of data collection: a multi-dimensional optimization over θ to fit the surrogate model, and a 2D optimization over $\mathbf{x}$ to identify the next sample point. To reduce the computational complexity in building the surrogate model, we adopt the public domain Netlab scaled conjugate gradient algorithm [Nabney, 2002] which runs in $O(|\mathcal{D}| \cdot |\theta|)$ time. While this algorithm avoids having to work with the Hessian explicitly, the active sample selection step requires the computation of the Hessian inverse, which takes $O(|\mathcal{D}| \cdot |\theta|^2 + |\theta|^3)$ time. To reduce the cost of optimization, we use a discrete lattice search or hill climbing, restricting our attention to locations over a uniform grid. If the number of locations on the grid is $|G|$, then each round of active mining, for either variance-based or entropy-based data collection, requires $O(|G| \cdot |\theta|^2 + |G|^2 \cdot |\theta|)$ time, plus the cost of computing the inverse Hessian.

Experimental Results

We now present empirical results demonstrating the effectiveness of our active mining strategy on both synthetic and real datasets. For evaluation, we consider classes of problems for which the 'right' answer is known, and pose questions such as: 'starting from an initial grid, how many samples does it take to mine the right number of higher-level structures?' The answer gives us an indication of how aggressive the sampling strategy is, its stability (i.e., once mined, does it continue to mine the patterns?), and comparisons with the other strategy. In this paper, we employed the Netlab suite of algorithms for GP modeling. Netlab supports a covariance formulation similar to Eq. 1, along with a bias term that overcomes our earlier assumption of zero mean. In addition, the model includes a noise term that can capture uncertainties in individual measurements; while this is not required for the deterministic functions considered here, it ensures that the numerical computation doesn't become unstable. All GP parameters are given a relatively broad Gaussian prior. A surrogate model was fit on a regularly spaced grid (more below), with a limit of 100 iterations for conjugate gradient search. The parameters for the SAL pocket finder were set as follows: 8-adjacency neighborhood graph (in step (b)), 0.75 cosine-angle for vector similarity (in step (c)), and 0.1 distance penalty metric in combining distance with direction (in step (d)). The standard variance-based sampling has no adjustable parameters; a fixed 8-adjacency neighborhood was utilized for defining $P(\mathbf{x})$ in entropy-based sampling. Optimization for Φ_V and Φ_E was conducted over the same grid as the domain of the surrogate function.

Synthetic Datasets

For the synthetic benchmark, we adopted the suite of test functions from [Gaviano et al., 2003], an ACM TOMS algorithm to readily generate classes of functions with known local and global minima. The algorithm systematically distorts a convex quadratic function with cubic or quintic polynomials to yield continuously differentiable (D-type) and twice continuously differentiable (D2-type) functions over the closed interval $[-1,1]^2$. Since our active mining proceeds by discrete search over a pre-defined grid, we evaluated the generated functions over a regular 21×21 grid in $[-1,1]^2$ ($|G| = 441$) and used these function values as the 'oracle' that is queried by the active mining mechanism. We verified whether in each instance, the SAL pocket finder is able to resolve all pockets when given a complete 21×21 dataset. This is necessary because the radii of the basins of attraction interact with the spacing of the sampling grid, and hence influence the number of samples available for aggregation by the SAL pocket finder. We found that the pocket finder is able to resolve only those generated functions that have up to 7 local minima; functions with more (e.g., 8–12) local minima use only a handful of points (typically 3–9) to represent some of their pockets, too few to be aggregated into a flow class under the SAL pocket finder's parameter settings. Hence, we pruned the automatically generated functions by requiring that that each local minimum have at least 12 samples per pocket, when sampled over the 21×21 grid. This yields a collection of 43 functions (21 D-type and 22 D2-type), with numbers of pockets ranging from 4 to 7. Fig. 7 depicts some of these functions.

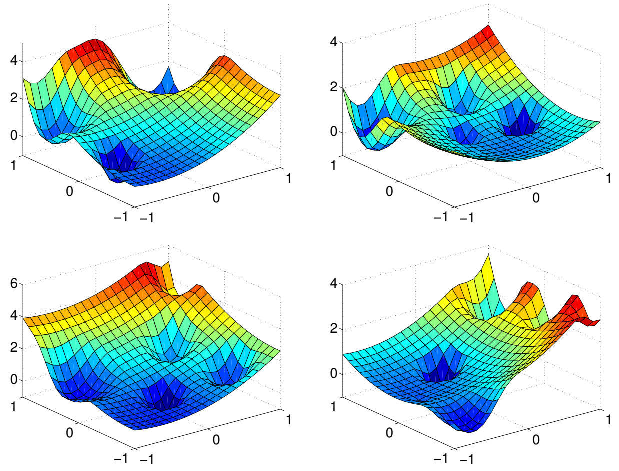

Figure 7: Example test functions with 4, 5, 6, and 7 pockets. Note that the viewpoint chosen makes visible only some of the pockets.

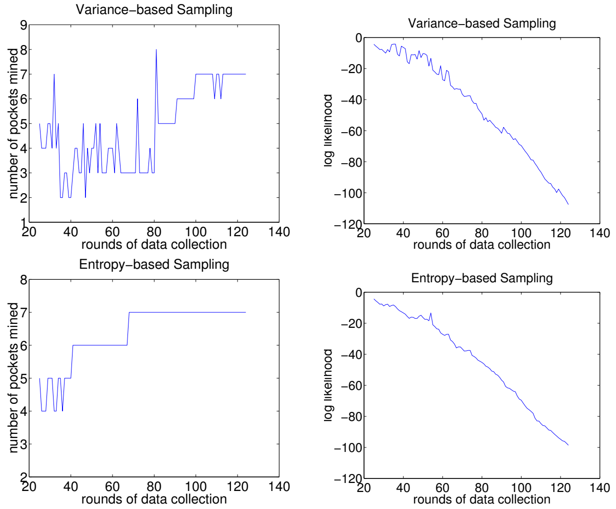

Figure 8: Pocket analysis performance on a 7-pocket function: (top) variance-based and (bottom) entropy-based sampling; (left) number of pockets found and (right) negative log-likelihood.

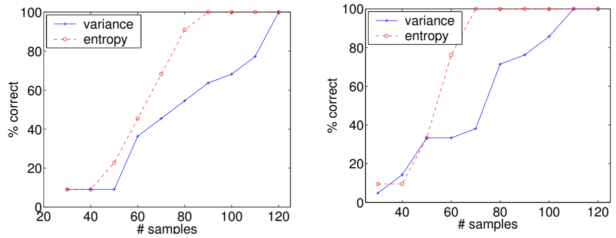

Figure 9: Overall pocket analysis performance (fraction of cases correctly identified) with increasing number of samples, for (left) D-type and (right) D2-type functions.

Both algorithms were initially seeded with a 5^2 design, comprising 25 points (about 5% of the design space of 441 points). Sampling was conducted for an additional 100 sample values (a total of 125 points, or about 25% of the design space). We reasoned that this is a good interval over which to monitor the performance of the sampling strategies, as even a regularly spaced grid covering 25% of the design space would mine the pockets correctly! Fig. 8 reveals the results for a 7-pocket function from our benchmark. Both sampling strategies systematically reduce the (negative) log likelihood (as estimated from the GP model parameters) but variance-based sampling shows more oscillatory behavior w.r.t. the number of pockets identified. On close inspection, we found that this strategy goes through stages where adjacent pockets are periodically re-grouped around sample values (which are mostly at the boundaries), causing rapid fluctuations in the SAL pocket finder's output. We say that this strategy is more prone to 'being surprised.' The number of pockets stabilizes around 7 only toward the end of the data collection interval. In contrast, the entropy-based sampling first identifies the seven pockets with 68 points, and proceeds to stabilize beyond this point. Similar results have been observed with other test functions.

Next, we analyzed the performance of both algorithms across all 43 test functions. We tested for what fraction of the datasets the analysis was correct by, and stayed correct following, a given number of rounds of sampling. Our hypothesis was that the D2-type functions, being smoother, are more easily modeled using GPs and should lend themselves to more aggressive sampling strategies. In addition, the entropy-based sampling strategy should be more effective w.r.t. number of rounds than the variance-based sampling. Fig. 9 shows that this is indeed the case.

Analysis of Wireless Configuration Spaces

Our second application involves characterization of configuration spaces of wireless system designs (see again Fig. 1). The goal is to understand the joint influence of selected configuration parameters on system performance. This can be achieved by identifying spatial aggregates in the configuration space, aggregating low level simulation data (typically multiple samples per configuration point) into regions of constrained shape. In particular, the setup in Fig. 1 is from a study designed to evaluate the performance of STTD (space-time transmit diversity) wireless systems, where the base station uses two transmitter antennas separated by a small distance, in an attempt to improve received signal strength. In this application, the aim is to assess how the power imbalance between the two branches impacts the performance (measured by bit error rate, BER) of the simulated system, across a range of signal-to-noise ratios (SNRs). When the signal components are significant compared to the noise components, and when the SNR ratios of the two branches are comparable, then it is well known that the system would yield high quality of BER performance. What is not so clear is how the performance will degrade as the SNRs move apart. Posed in the spatial aggregation framework, this objective translates into identifying and characterizing (in terms of width, or power imbalance) the pocket in the central portion of the configuration space. Identifying and characterizing other pockets is not as important, since some of them will actually contain suboptimal configurations.

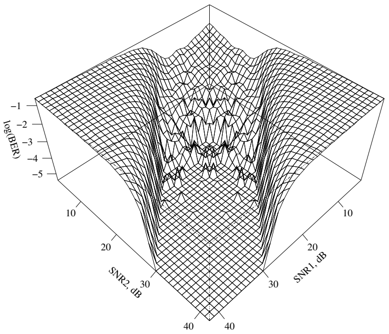

Figure 10: Estimates of BER performance in a space of wireless system configurations.

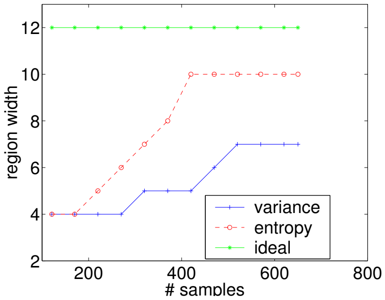

Figure 11: Performance of active mining strategies on wireless simulation data, characterizing width of the main pocket in Fig. 10 with increasing numbers of samples.

We adopt an experimental methodology similar to that in the previous case studies, and created an 'oracle' from the simulation data described in [Verstak et al., 2002]. Fig. 10 demonstrates that the dataset is quite noisy, especially when the SNR values are low. The design of the oracle, surrogate model building, and sample selection all employ a 55 × 55 grid over the configuration space (SNR levels ranging from 3dB to 58dB for each antenna). Both variance-based sampling and entropy-based sampling were initialized using a 11^2 design (about 4% of the configuration space). Sampling was conducted for an additional 650 points, yielding a total of 771 points (25% of the design space, as with the earlier studies). For each round of active mining, we determined the majority class occupied by points having equal SNR and determined the maximum width of this class. This measure was periodically tracked across the rounds of data collection. Fig. 11 shows how the sampling strategies fare compared to the correct estimate of 12dB, as reported in [Verstak et al., 2002] by applying an analysis algorithm over the entire dataset. Entropy-based sampling once again selects data that systematically clarify the nature of the pockets, and cause a progressive widening of the trough in the middle. However, it doesn't identify the ideal width of 12dB (within the given samples). We reason that this is because the GP model has difficulty approximating the steep edge of the basin. Variance-based sampling fares worse and demonstrates a slower growth of width across samples. This application highlights the utility of our framework for analyzing both qualitative and quantitative properties of spatial aggregates. This work thus helps improve the efficiency of wireless system characterization by mining the essential attributes of good designs using an economy of simulations.

Discussion

This paper has presented a novel integration of approaches from three areas, namely qualitative spatial reasoning, probabilistic modeling using GPs, and active data mining. The marriage of Gaussian processes and SAL is a natural one at many levels. First, by focusing on covariance structures rather than value approximation, the modeling of qualitative characteristics via GPs mirrors the mining of spatial aggregates in SAL. Both GPs and SAL share the traits of locality of computations, weak priors on data characteristics, and capability to model a wide range of high-level phenomena. Second, the sound statistical basis of GPs supports a rigorous way to integrate qualitative and quantitative modeling of spatial datasets. Finally, the multi-level nature of SAL is elegantly captured in the compositional structure of covariance functions. Earlier work [Bailey-Kellogg and Ramakrishnan, 2001] used ambiguity in SAL's computations to formulate indicator random variables that focus sampling at given points, but sample selection was driven by the quality of function approximation. Subsequently [Ramakrishnan and Bailey-Kellogg, 2003], we focused on how GPs can model SAL's localized computations, but did not integrate the two in order to perform active mining. This paper represents a logical culmination of this research and demonstrates a mechanism that makes judicious use of limited data by combining the strengths of GPs (for spatial modeling) with SAL (for identifying high-level structures).

There are several possible extensions to the work presented here. First, our assumption of sampling over a defined grid can be relaxed and the scope of active mining can be expanded to include subsampling. Second, the modeling of vector fields using GPs warrants further investigation, in particular to address the issue of how to model data fields given only (or also) derivative information or when the underlying function is not smooth or differentiable. Other investigators have done related work in this area [Cornford et al., 1998]. Third, we assume here that the model (of flow classes) posited by SAL is correct, and use this information to drive the sampling. To overcome this assumption, we must create a probabilistic model of SAL's computations (including uncertainty and non-determinism in aggregation procedures) and integrate this model with the GP model for the data fields. These and similar ideas will help establish the many ways in which mathematical models of data approximation can be integrated with qualitative analysis algorithms.

Acknowledgements

This work is supported in part by US NSF grants IIS-0237654, EIA-9984317, and IBN-0219332.

References

[Bailey-Kellogg and Ramakrishnan, 2001] Bailey-Kellogg, C. and Ramakrishnan, N. (2001). Ambiguity-Directed Sampling for Qualitative Analysis of Sparse Data from Spatially Distributed Physical Systems. In *Proc. IJCAI*, pages 43–50.

[Bailey-Kellogg and Ramakrishnan, 2004] Bailey-Kellogg, C. and Ramakrishnan, N. (2004). Spatial Aggregation for Qualitative Assessment of Scientific Computations. In *Proc. AAAI*. to appear.

[Bailey-Kellogg and Zhao, 2001] 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 et al., 1996] Bailey-Kellogg, C., Zhao, F., and Yip, K. (1996). Spatial Aggregation: Language and Applications. In *Proc. AAAI*, pages 517–522.

[Cornford et al., 1998] Cornford, D., Nabney, I., and Williams, C. (1998). Adding Constrained Discontinuities to Gaussian Process Models of Wind Fields. In *Proceedings of NIPS*, pages 861–867.

[Gaviano et al., 2003] Gaviano, M., Kvasov, D., Lera, D., and Sergeyev, Y. (2003). Algorithm 829: Software for Generation of Classes of Test Functions with Known Local and Global Minima for Global Optimization. *ACM Transactions on Mathematical Software*, Vol. 29(4):pages 469–480.

[Journel and Huijbregts, 1992] Journel, A. and Huijbregts, C. (1992). *Mining Geostatistics*. Academic Press, New York.

[MacKay, 1992] MacKay, D. (1992). Information-Based Objective Functions for Active Data Selection. *Neural Computation*, Vol. 4(4):pages 590–604.

[Nabney, 2002] Nabney, I. (2002). *Netlab: Algorithms for Pattern Recognition*. Springer-Verlag.

[Neal, 1997] Neal, R. (1997). Monte Carlo Implementations of Gaussian Process Models for Bayesian Regression and Classification. Technical Report 9702, Department of Statistics, University of Toronto.

[Ordóñez and Zhao, 2000] Ordóñez, I. and Zhao, F. (2000). STA: Spatio-Temporal Aggregation with Applications to Analysis of Diffusion-Reaction Phenomena. In *Proc. AAAI*, pages 517–523.

[Ramakrishnan and Bailey-Kellogg, 2002] Ramakrishnan, N. and Bailey-Kellogg, C. (2002). Sampling Strategies for Mining in Data-Scarce Domains. *IEEE/AIP CiSE*, Vol. 4(4):pages 31–43.

[Ramakrishnan and Bailey-Kellogg, 2003] Ramakrishnan, N. and Bailey-Kellogg, C. (2003). Gaussian Process Models of Spatial Aggregation Algorithms. In *Proc. IJCAI*, pages 1045–1051.

[Verstak et al., 2002] Verstak et al., A. (2002). Using Hierarchical Data Mining to Characterize Performance of Wireless System Configurations. Technical Report cs.CE/0208040, CoRR.

[Williams, 1998] Williams, C. (1998). Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In Jordan, M., editor, *Learning in Graphical Models*, pages 599–621. MIT Press, Cambridge, MA.