Gaussian Process Models of Spatial Aggregation Algorithms

Naren Ramakrishnan
Department of Computer Science
Virginia Tech, VA 24061, USA
naren@cs.vt.edu

Chris Bailey-Kellogg
Department of Computer Sciences
Purdue University, IN 47907, USA
cbk@cs.purdue.edu

Abstract

Multi-level spatial aggregates are important for data mining in a variety of scientific and engineering applications, from analysis of weather data (aggregating temperature and pressure data into ridges and fronts) to performance analysis of wireless systems (aggregating simulation results into configuration space regions exhibiting particular performance characteristics). In many of these applications, data collection is expensive and time consuming, so effort must be focused on gathering samples at locations that will be most important for the analysis. This requires that we be able to functionally model a data mining algorithm in order to assess the impact of potential samples on the mining of suitable spatial aggregates. This paper describes a novel Gaussian process approach to modeling multi-layer spatial aggregation algorithms, and demonstrates the ability of the resulting models to capture the essential underlying qualitative behaviors of the algorithms. By helping cast classical spatial aggregation algorithms in a rigorous quantitative framework, the Gaussian process models support diverse uses such as directed sampling, characterizing the sensitivity of a mining algorithm to particular parameters, and understanding how variations in input data fields percolate up through a spatial aggregation hierarchy.

1 Introduction

Many important tasks in data mining, scientific computing, and qualitative modeling involve the successive and systematic spatial aggregation and redescription of data into higher-level objects. For instance, consider the characterization of WCDMA (wideband code-division multiple access) wireless system configurations for a given indoor environment. In noisy channels, the performance goal is to quantitatively assess the relationship between the signal-to-noise ratio (SNR) and the bit error rate (BER) or bit error probability (BEP) of the realized configuration. To improve performance in office environments (characterized by doorways, walls, cubicles), a common trick used is to incorporate space-time transmit diversity (STTD). Instead of a single transmitter antenna, the base station uses two transmitter antennas separated by a small distance. If the signal from one of the antennas is weak, the signal from another is likely to be high, and the overall performance is expected to improve. In this application, it is important to assess how the power-imbalance between the two branches impacts the BEP of the simulated system, across a range of SNRs (see Fig. 1; [Verstak *et al.*, 2002]).

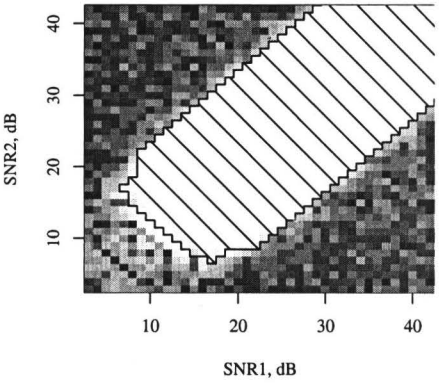
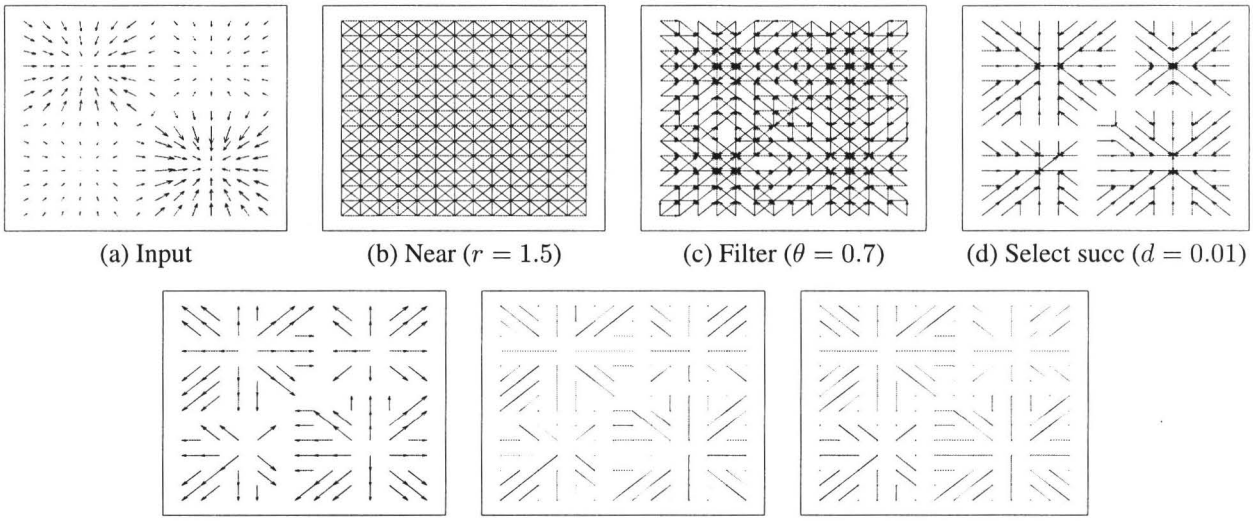


Figure 1: Mining configuration spaces in wireless system configurations. The shaded region denotes the largest portion of the configuration space where we can claim, with confidence at least 99%, that the average bit error probability (BEP) is acceptable for voice-based system usage. Each cell in the plot is the result of the spatial and temporal aggregation of hundreds of time-consuming wireless system simulations.

Characterizing the performance of WCDMA systems requires the identification of multi-level spatial aggregates in the high-dimensional configuration spaces of wireless systems. The lowest level (input) contains individual Monte Carlo simulation runs providing unbiased estimates of BEPs. This space is high-dimensional (e.g. ≥ 10), owing to the multitude of wireless system parameters (e.g. channel models, fading characteristics, coding configurations, and hardware controls). Wireless design engineers prefer to work in at most two or three dimensions (e.g. to study the effect of power imbalance on system performance) for ease of tunability and deployment. The next level of spatial aggregation thus contains *buckets* which aggregate data in terms of two dimensions, using various consistency constraints and design specifications. Finally, the third level aggregates buckets into *regions* of con-



(a) Input (b) Near ($r = 1.5$) (c) Filter ($\theta = 0.7$) (d) Select succ ($d = 0.01$)

(e) Select pred ($d = 0.01$) (f) Redescribe (g) Bundle

Figure 2: Key steps in vector field analysis.

strained shape; the shape of the regions illustrates the nature of joint influence of the two selected configuration parameters on performance. Specific region attributes, such as width, provide estimates for the thresholds of sensitivity of configurations to variations in parameter values.

The results of such mining are important for both qualitative and quantitative analysis. For instance, when the average SNRs of the two branches are equal, the BEP is minimal and the width of the mined region in Fig. 1 depicts the largest acceptable power imbalance (in this case, approximately 12dB). However, the width is not uniform and the region is narrower for smaller values of the SNRs. The qualitative result is that system designs situated in the lower left corner of the configuration space are more sensitive to power imbalance in the two branches.

Each input data point captures the results of a wireless system simulation which takes hours or even days (the simulations in Fig. 1 were conducted on a 200-node Beowulf cluster of workstations). Thus it is imperative that we focus data collection in only those regions that are most important to support our data mining objective, viz. to qualitatively assess the performance in configuration spaces. This requires that we model the functioning of the data mining algorithm, in order to optimize sample selection for utility of anticipated results. Modeling data mining algorithms in this manner is useful for closing the loop, characterizing the effects of the data mining algorithm's parameters, and improving our understanding of how variations in data fields percolate up through the layers. A particularly interesting application is to use such modeled structures to design information-theoretic measures for evaluating experimental designs [MacKay, 1992] and for active data selection [Cohn *et al.*, 1996; Denzler and Brown, 2002].

In order to address these goals, this paper develops a novel Gaussian process approach to modeling algorithms that mine spatial aggregates. We first overview the Spatial Aggregation mechanism for spatial data mining and the Gaussian process approach to Bayesian modeling. We then show how to integrate the two approaches in order to achieve our goal of probabilistically modeling spatial data mining algorithms. We illustrate this ability within the context of identifying pockets underlying the gradient in a field — an application that captures many of the interesting characteristics of more complex studies like the wireless application.

2 Spatial Aggregation

The Spatial Aggregation Language (SAL) [Bailey-Kellogg *et al.*, 1996; Yip and Zhao, 1996], provides a set of operators and data types, parameterized by domain-specific knowledge, for uncovering and manipulating multi-layer geometric and topological structures in spatially distributed data. SAL applications construct increasingly abstract descriptions of the input data by utilizing knowledge of physical properties such as continuity and locality, expressed with the vocabulary of metrics, adjacency relations, and equivalence predicates. To understand the SAL approach (see Fig. 2), consider a SAL program for analyzing flows in a vector field (e.g. wind velocity or temperature gradient).

In the first level, the goal is to group input vectors (a) into paths so that each sample point has at most one predecessor and at most one successor. SAL breaks the process into two key steps, one capturing locality in the domain space (i.e. sample location), and the other capturing similarity in the feature space (i.e. vector direction). A neighborhood graph aggregates objects with a specified adjacency predicate expressing the notion of locality appropriate for a given domain. As shown (b), a sample point's neighbors include all other points within some specified radius r. Feature comparison then must consider only neighbors in this graph, thereby exploiting physical knowledge to gain computational efficiency while maintaining correctness. Here we break feature comparison into a sequence of predicates and graph operations. In particular, we first filter the graph (c), applying a predicate that keeps only those edges whose direction is similar enough

(within some angle tolerance θ) to the directions of the vectors at the endpoints. The remaining graph has some "junction" points where vector direction suggests multiple possible neighbors, and the most appropriate path extension from the point must be chosen. A similarity metric sums the distance between the junction and a neighbor, weighted by a constant d, and the difference in vector direction at the junction and the neighbor. The most similar neighbor for the junction is selected ((d) and (e), for successor and predecessor junctions, respectively).

The remaining graph edges are collected and redescribed as more abstract streamline curve objects (f), for the second level of analysis. Again, computation is localized so that only neighboring streamlines are compared. The neighborhood graph here (not shown) uses an adjacency predicate that declares streamlines neighbors if their constituent points were in the first level. It is then straightforward to identify convergent flows (g) with an equivalence predicate that tests when constituent points form a junction in the graph in (c). If desired, these flow bundles can be abstracted and analyzed at an even higher level.

SAL's uniform spatial reasoning mechanism, instantiated with appropriate domain knowledge, has proved successful in applications ranging from decentralized control design [Bailey-Kellogg and Zhao, 1999; 2001], to weather data analysis [Huang and Zhao, 1999], to analysis of diffusion-reaction morphogenesis [Ordóñez and Zhao, 2000]. Recent work has focused on optimizing sample selection for applications where data collection is expensive, including identifying flows in multi-dimensional gradient fields [Bailey-Kellogg and Ramakrishnan, 2001] and analyzing matrix properties via perturbation sampling [Ramakrishnan and Bailey-Kellogg, 2002]. This paper provides the mathematical foundations necessary for the modeling of such SAL programs to support the meta-level reasoning tasks outlined in the introduction.

3 Gaussian Processes

Gaussian processes have become popular in the last few years, especially as a unifying framework for studying multivariate regression [Rasmussen, 1996], pattern classification [Williams and Barber, 1998], and hierarchical modeling [Menzefricke, 2000]. The underlying idea can be traced back to the geostatistics technique called *kriging* [Journel and Huijbregts, 1992], named after the South African miner Danie Krige. In kriging, the unknown function to be modeled (e.g., ozone concentration) over a (typically) 2D spatial field is expressed as the realization of a stochastic process. A prior is placed over the function space represented by this stochastic process, by suitably selecting a covariance function. Given measured function values at sample locations, kriging then proceeds to estimate the parameters of the covariance function (and any others pertaining to the random process). Using such values a prediction of the response variable can then be made for a new sample point, typically using MAP or ML inference. This basic approach is still popular in many tasks of spatial data analysis.

Even though parameters are estimated in this approach, it is important to note that kriging is fundamentally a memory-based technique, since the estimated parameters only describe the underlying covariance function of a stochastic process. Thus, predictions of the response variable for new sample points are conditionally dependent on the measured values and *their* sample points; by unrolling the effect of the parameters of the random process, we can directly express this dependency.

Kriging is often motivated as a local modeling technique, capable of approximating or interpolating functions with multiple local extrema, and generalizes well to applications exhibiting anisotropies and trends. The stochastic prior is also viewed as a mathematically elegant mechanism to impart any available domain knowledge to the modeling technique. In 1989, Sacks et al. [Sacks *et al.*, 1989] showed how kriging can actually be used to model processes with *deterministic* outcomes, especially in the context of computer experiments. The justification for modeling a deterministic code as a stochastic process is often that even though the response variable is deterministic, it may 'resemble the sample path of a suitably chosen stochastic process' [Sacks *et al.*, 1989]. Alternatively, using a stochastic process prior can be viewed as a Bayesian approach to data analysis [Sivia, 1996], and this is the idea emphasized by most recent computer science research in Gaussian processes [Gibbs, 1997; Rasmussen, 1996]. The stochastic process can be suitably formulated to ensure that the model reproduces the same response value for repeated invocations of a given sample input (i.e., absence of random error). For instance, the Gaussian prior can be chosen so that the diagonal entries of the covariance matrix are 1, meaning that the model should interpolate the data points.

In the recent past, Gaussian processes have become popular in the statistical pattern recognition community [MacKay, 1997] and graphical models literature [Jordan (ed.), 1998]. Neal established the connection between Gaussian processes and neural networks with an infinite number of hidden units [Neal, 1996]. Such relationships allow us to take traditional learning techniques and re-express them as imposing a particular covariance structure on the joint distribution of inputs. For instance, we can take a trained neural network and mine the covariance structure implied by the weights (given mild assumptions such as a Gaussian prior over the weight space). Williams motivates the usefulness of such studies and describes common covariance functions [Williams, 1998].

Williams and Barber [Williams and Barber, 1998] describe how the Gaussian process framework can be extended to classification, where the modeled variable is categorical. Essentially, the idea is to (i) use a logistic function to conduct traditional Gaussian regression modeling, and (ii) adopt a *softmax* function to bin the logistic output into a given set of classes. This means that the logistic function uses a "latent variable" as input in its computation, since its values are not provided by the dataset.

4 Gaussian Processes for Spatial Aggregation

SAL programs construct multi-layer spatial aggregates based on specified local adjacency relations, similarity metrics, and consistency checks. We describe here how to capture the

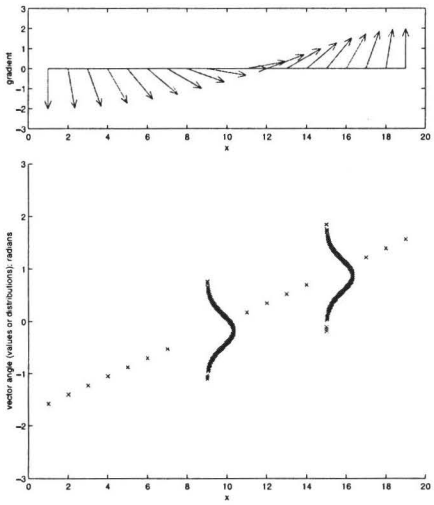


Figure 3: Modeling the reversal of gradients in a 1D field using Gaussian processes. (top) Original field. (bottom) Given measured values of gradient vector angles at specific data points (blue), the model posits that the conditional distribution of the angle at unseen data points is a Gaussian (shown in red).

qualitative behaviors of such aggregates using Gaussian processes. The essence of a Gaussian process is its covariance structure, so we focus on determining covariance structures in a SAL program. For example, in the two-layer SAL program of Sec. 2, the parameters (r, θ, d) impose a covariance structure by specifying the reach of the neighborhood graph, enforcing the similarity of angles in the vector field, and penalizing for the distance at decisions involving junctions.

4.1 Covariance Structure

We now describe how to model the covariance structure of a given SAL program. We give the mathematical framework for the case of mining a 1D field to determine if there is a reversal of gradient as we move along the spatial dimension, but essentially the same machinery applies to two and higher dimensional spaces. The basic problem is one of classifying 1D points to determine the qualitative structure of same-direction flows. Fig. 3 (top) depicts the given input field along the x dimension. As is shown, the field consists of unit vectors with different orientation. The Gaussian process approach is first to model an underlying regressed variable and then to use a logistic or softmax function to bin the output into classes. In our application, the regressed variable represents the gradient and can be simply summarized as the angle of unit vector orientation y in Fig. 3 (top). In other applications, the regressed variable could be an unobserved 'latent' variable. In either case, it is modeled as a function f of the input x.

First, assume f to be a Gaussian process on x, meaning that the conditional probability distribution of y given a value of x is a Gaussian. For instance, Fig. 3 (bottom) depicts measured values of y superposed with distributions of y at two unseen points. A covariance structure among the y values could, for example, capture the intuition that adjacent values of y should agree more than distant values. The goal of modeling is to determine the extent and stringency of this neighborhood relation — one of the defining parameters of a SAL program. Specifically, we posit a process such as:

$$f(x) = \alpha + Z(x) \tag{1}$$

The idea then is to estimate a model f' of the same form as f, on the basis of a given set of k observations $\{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$. A typical choice for Z in f' is a random process with zero mean and covariance $\sigma^2 R$, where scalar σ^2 is the estimated variance and R is a matrix that captures the correlation between the inputs (i.e., the given locations). Notice that even though the input is one dimension, the size of R depends on the number of locations for which gradient measurements are available. The above model for f also includes the constant term α; this can be estimated based on the k observations, or we can substitute more complex terms (e.g. linear), or even omit it altogether.

The functional form of R (including its parameterization) in effect defines the stochastic process and must be carefully chosen to reflect the underlying data's fidelity or any domain-specific assumptions about local variation. The parameters of the process are then estimated using multidimensional optimization involving a suitable objective function. For instance, given the following form for R:

$$R(x_i, x_j) = e^{-\rho|x_i - x_j|^2} \tag{2}$$

the problem reduces to estimating ρ from the given data. Notice that this formulation for R implicitly enforces that the model exactly interpolate the given data points, since $R(x_i, x_i) = 1$. A common objective function for estimating ρ is to minimize the mean squared error (MSE), $E\{(f' - f)^2\}$ between f and f'. The ρ that minimizes MSE is given by the solution to the optimization problem:

$$\max_{\rho} \left(-\frac{k}{2}(\ln \sigma^2 + \ln |\mathbf{R}|) \right) \tag{3}$$

where $\mathbf{R}$ is the symmetric correlation matrix formed from R. For a new sample point x_{k+1}, a prediction for the regressed variable is given by:

$$f'(x_{k+1}) = \hat{\alpha} + \mathbf{r}^T(x_{k+1})\mathbf{R}^{-1}(\mathbf{y} - \hat{\alpha}\mathbf{I_k}) \tag{4}$$

where $\mathbf{r}$ is the correlation vector between the response at x_{k+1} and all the other k points (derived from R), $\mathbf{I_k}$ is the identity vector of dimension k, and $\hat{\alpha}$ is the estimate of α given by:

$$\hat{\alpha} = (\mathbf{I_k^T}\mathbf{R}^{-1}\mathbf{I_k})^{-1}\mathbf{I_k^T}\mathbf{R}^{-1}\mathbf{y} \tag{5}$$

The variance in the estimate is given by:

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \hat{\alpha}\mathbf{I_k})^T\mathbf{R}^{-1}(\mathbf{y} - \hat{\alpha}\mathbf{I_k})}{k} \tag{6}$$

In this case, the optimization is one-dimensional due to the presence of the single parameter ρ. With a different parameterization, we will employ multi-dimensional optimization over the entire set of hyperparameters. When dimensionality is large, the hyperparameters are estimated using MCMC methods. Once such a modeling is complete, as discussed in the previous section, we can relate a categorical class variable

to y using softmax functions. For instance, the reversal of the gradient in Fig. 3 can be captured by first using the Gaussian process model to make predictions of the gradient at untested points and then determining if (and where) a zero crossing occurs.

The above equations extend naturally to a 2D case such as that described in Sec. 2. The covariance prior has to be suitably parameterized and we also have the option of taking into account any interactions between the two dimensions (both linear and nonlinear).

4.2 Modeling Many Layers

When SAL programs consist of many layers, we need to develop a sequence of Gaussian process models, each with a suitable covariance function, which can then be superposed to yield a composite covariance function. Recall that while one could simply assess the covariance of the output field for sample values of the parameters and a given input field, the real purpose of a Gaussian process model is to express the covariance of the output *as a function of the characteristics of the input*. This is the key property that allows reasoning about closing the loop and selecting optimal samples. In addition, Gaussian process models help capture the randomness inherent in some of SAL's computations, e.g. non-determinism in labeling, and variations due to how ties are broken for aggregation purposes. Refer again to Sec. 2 for an example of the types of operations that the covariance model must capture.

At the very bottom of the hierarchy is the input data field. For applications characterized by expensive data collection (as in the introduction), it can be advantageous to start with a *sparse* set of sample data. The Gaussian modeling approach to regression is ideal for creating *surrogate* representations of data fields from such a sparse dataset. That is, given a sparse set of samples, interpolate a dense field satisfying those values and incorporating any appropriate domain knowledge, as discussed above regarding kriging. Such surrogate functions can then been used as the starting points for qualitative analysis [Bailey-Kellogg and Ramakrishnan, 2001].

The operators in a SAL level deal with both locality (which object locations are close to which other ones, as encapsulated in a neighborhood graph) and similarity (which object features are close to which other ones, as encapsulated in metrics and predicates). For instance, in the example of Sec. 2, two points are assigned to the same pocket if they are spatially proximate and their flows converge. Here the Gaussian process is classification (or more generally, density estimation). A popular covariance structure for an n-dimensional input field captures locality:

$$R(x^{(k)}, x^{(l)}) = \zeta \prod_{i=1}^{n} e^{-\rho_i |x_i^{(k)} - x_i^{(l)}|^\eta} \tag{7}$$

where the expression relates the function values at positions $x^{(k)}$ and $x^{(l)}$. If $\eta \in [0, 2]$, then the covariance function will be positive definite, satisfying the normalization constraints of *a posteriori* inference.

To see how to capture similarity, consider when two sample locations are classified into the same trajectory in Sec. 2. In addition to being spatially proximate (as inferred by SAL's neighborhood calculations), the underlying vector fields must also be similar in direction. Expressing the covariance in terms of position alone can cause the resulting estimated hyperparameters to be misleading or difficult to interpret, as their effect is *confounded* with the underlying vector field. One solution is to artificially inflate the dimensionality, so that position and direction *together* describe the data. Besides increasing the dimensionality, this approach spells trouble for estimation using MCMC methods since significant portions of the sample space will remain unsampled and it would be difficult to assess their effects on the minimized functional. An alternative solution is to use the fact that the vector field is itself a surrogate and add a term to the covariance *outside the above structure*, capturing the contribution due to similarity in the vector field. We place a Gamma prior on this term with a shape parameter that ensures that its role is secondary to the covariance structure on position (directional similarity alone is not enough for high covariance at the output; the sample locations also must be spatially proximate). This is recognized in the statistics community as a hierarchical prior and described in detail in [Neal, 1997].

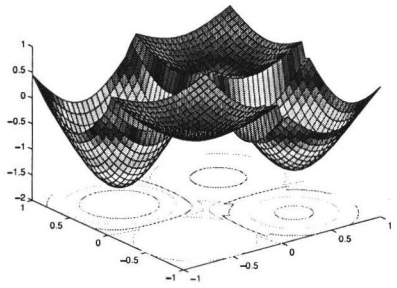


Figure 4: A 2D pocket function.

5 Experimental Results

In order to test our approach, we studied de Boor's pocket function (see Fig. 4):

$$\begin{aligned} \alpha(\mathbf{X}) &= \cos\left(\sum_{i=1}^{n} 2^i \left(1 + \frac{x_i}{|\, x_i \,|}\right)\right) - 2 \\ \delta(\mathbf{X}) &= \|\mathbf{X} - 0.5\mathbf{I}\| \\ p(\mathbf{X}) &= \alpha(\mathbf{X})(1 - \delta^2(\mathbf{X})(3 - 2\delta(\mathbf{X}))) + 1 \end{aligned}$$

where $\mathbf{X}$ is the n-dimensional point $(x_1, x_2, \cdots, x_n)$ at which the pocket function p is evaluated, $\mathbf{I}$ is the identity n-vector, and $\|\cdot\|$ is the L_2 norm. This function exploits the fact that the volume of a high dimensional cube is concentrated in its corners and p is designed so that it has a "dip" in each corner. It embodies many aspects of datasets like those encountered in the wireless simulation study, including multiple local extrema, non-systematic variation in the location of the pockets, and regional variation. The pocket function is also important as a benchmark for high-dimensional data exploration, where the goal is to identify the most interesting regions of the design space without necessarily conducting a (costly) global optimization over the entire design space. Data mining programs are hence required to identify the most promising regions using as few function evaluations as possible.

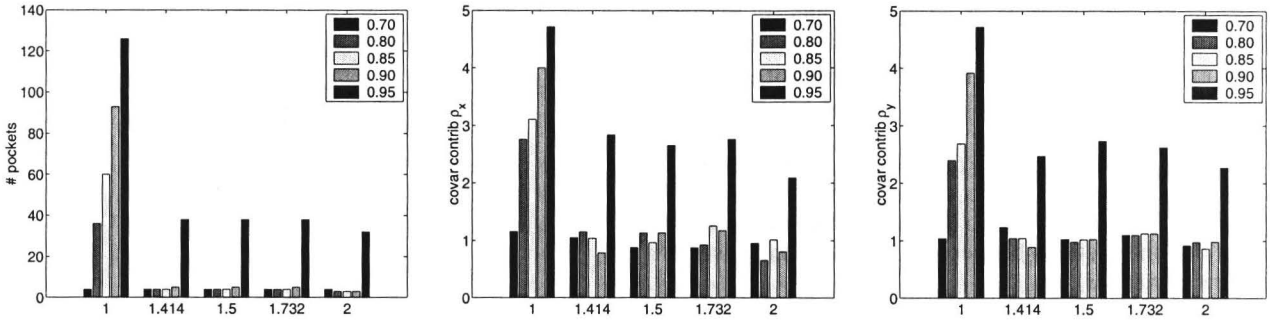


Figure 5: Modeling a SAL program to mine pockets in gradient fields. (a) Variation in number of pockets mined by the SAL program for various values of (r, θ). (b) Covariance contribution in x dimension for various values of (r, θ). (c) Covariance contribution in y dimension for various values of (r, θ). In all charts, r varies by group and θ varies within group.

A SAL program to identify the number of pockets starts with some samples of the pocket function. The lowest level of modeling involves a kriging interpolation over a uniform grid (we chose size 13^n for testing). Then the approach of Sec. 2 is applied to the gradient vector field of this scalar field: the second level bundles points into curves, and the third aggregates these into flows. Each convergent flow represents one pocket. One could mine the covariance structures for each layer separately; we unfold these mappings here to obtain a single covariance structure summarizing all three layers. This is because the structure (esp. the contributions of each dimension) is easiest to interpret in terms of the original spatial field.

We conducted a parameter sweep over (r, θ, d) as:

$$\begin{aligned} r &\in \{1, \sqrt{2}, 1.5, \sqrt{3}, 2\} \\ \theta &\in \{0.7, 0.8, 0.85, 0.9, 0.95\} \\ d &\in \{0.01, 0.02, 0.03, 0.04, 0.05\} \end{aligned}$$

and used Neal's Bayesian modeling software [Neal, 1997] to construct Gaussian process classifiers for the flow classes. Covariance contributions in the ρ terms (Eq. 7) from both the dimensions was estimated using hybrid Monte Carlo (aggressive schemes to evolve the system state by adding higher order terms). This procedure uses a leapfrog scheme to suppress random walk behavior by selective iteration between Gibbs sampling scans and latent value updates.

Our results indicated a strong positive correlation between the x and y covariance contributions, bringing out the symmetry in the underlying SAL computations. The number of pockets mined was constant across the values of d (other parameters fixed), and one of the goals of our study was to determine if this negligible effect of d is captured in the covariance structures. (The effect of d would actually be more pronounced in other spatial fields but not so much in the pocket function due to the inherent symmetry.) Parameters r and θ produced the most variation in the covariance contributions with $\theta = 0.95$ causing an abrupt jump in the number of mined pockets. This is due to the rather stringent limit imposed on vector similarity arising from the nonlinearity of the cosine metric. Fig. 5 summarizes the results for a 2D pocket function, where we have averaged the covariance contributions across all ds, for given r and θ.

As the number of pockets increases (Fig. 5(a)), the covariance contributions increase (Fig. 5(b,c)) approximately quadratically. In other words, as the underlying latent function varies rapidly along the given dimensions, we cannot stray "too far" away from a given sample point when making predictions at test points. The reciprocal of the covariance scale term is often referred to the *characteristic length* of a dimension. This gives an estimate of "how far" a given dimension's effect holds. When only four pockets are mined, the characteristic length is about 1, meaning pockets occupy a width of 1×1 (exactly one fourth of the total space). As more pockets are mined, the characteristic length drops to about 0.4. It is also interesting to note that the abrupt jump in the number of pockets for $\theta = 0.95$ is reflected by a similar increase in the covariance contributions for this value. Essentially, vector and edge directions have to be so similar that few long "runs" can be aggregated as streamlines. This brings out the capability of the Gaussian process approach to capture the essentials of a spatial aggregation algorithm.

6 Discussion

This paper has demonstrated a novel Gaussian process approach to modeling the qualitative behavior of SAL programs; in contrast to much of the literature where Gaussian processes are used for pattern classification and regression [Rasmussen, 1996; Gibbs, 1997], our work takes *existing* data mining algorithms and recasts them in terms of Gaussian priors. To the best of our knowledge, this is the first study to completely model a qualitative data mining algorithm in terms of a process framework, summarizing the transformation from data to higher-level aggregates. This is an important step in firmly establishing a probabilistic basis for spatial aggregation computations. The modeling undertaken here, while expensive, is justifiable for studies such as the wireless system characterization described in the introduction.

There are several immediate gains from the work presented here; due to space limitations we only mention them briefly. First, the Gaussian process model can characterize experimental design criteria such as entropy as a functional w.r.t. the input space, allowing us to use the mined covariance structure to focus sampling at the most informative points (e.g., see [Bailey-Kellogg and Ramakrishnan, 2001]). It is important to note, however, that the approach taken in [Bailey-Kellogg and Ramakrishnan, 2001] only addresses the lowest

levels of a hierarchy and is unable to reason about higher-level, more abstract processes of redescription and aggregation as is done here. Second, Gaussian process models allow us to study the effects of different SAL parameters for a given class of datasets, e.g. the inference above of the negligible role of d in the mining process. Finally, it allows us to take algorithms that function in differing ways (and using different sets of parameters) and places them on a common footing, namely the language of covariance structures. This means that we can reason about the applicability of different algorithms by studying the constraints they impose on spatial locality and field similarity.

Gaussian processes have recently been linked to kernel-based methods, as used in support vector machines [Cristianini and Shawe-Taylor, 2000]; we intend to explore this connection in future work. Kernel-based methods are attractive in their promise to overcome the curse of dimensionality by the use of nonlinear projections, a facet that is of critical importance for mining data from large parameter sweeps. As the need for data mining in computational science gains prominence, process models will be crucial to achieve effective utilization of data for mining purposes.

Acknowledgements

The authors thank Feng Zhao and Layne Watson for helpful comments. This work is supported by US NSF grants EIA-9974956, EIA-9984317, and EIA-0103660.

References

[Bailey-Kellogg and Ramakrishnan, 2001] C. Bailey-Kellogg and N. Ramakrishnan. 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)*, pages 43–50, 2001.

[Bailey-Kellogg and Zhao, 1999] C. Bailey-Kellogg and F. Zhao. Influence-Based Model Decomposition. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, pages 402–409, 1999.

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

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

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

[Cristianini and Shawe-Taylor, 2000] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[Denzler and Brown, 2002] J. Denzler and C.M. Brown. Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24(2):pages 145–157, Feb 2002.

[Gibbs, 1997] M.N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1997.

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

[Jordan (ed.), 1998] M.I. Jordan (ed.). *Learning in Graphical Models*. MIT Press, 1998.

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

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

[MacKay, 1997] D.J. MacKay. Gaussian Processes: A Replacement for Supervised Neural Networks? In *Lecture Notes of Tutorial at Neural Information Processing Systems (NIPS'97)*, 1997.

[Menzefricke, 2000] U. Menzefricke. Hierarchical Modeling with Gaussian Processes. *Communications in Statistics*, Vol. 29(4):pages 1089–1108, 2000.

[Neal, 1996] R.M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, NY, 1996. Lecture Notes in Statistics No. 118.

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

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

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

[Rasmussen, 1996] C.E. Rasmussen. *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*. PhD thesis, University of Toronto, 1996.

[Sacks *et al.*, 1989] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, Vol. 4(4):pages 409–435, 1989.

[Sivia, 1996] D.S. Sivia. *Data Analysis: A Bayesian Tutorial*. Oxford University Press, 1996.

[Verstak *et al.*, 2002] A. Verstak, N. Ramakrishnan, K.K. Bae, W.H. Tranter, L.T. Watson, J. He, C.A. Shaffer, and T.S. Rappaport. Using Hierarchical Data Mining to Characterize Performance of Wireless System Configurations. Technical Report cs.CE/0208040, Computing Research Repository, Aug 2002.

[Williams and Barber, 1998] C.K.I. Williams and D. Barber. Bayesian Classification with Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20(12):pages 1342–1351, Dec 1998.

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

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