SAND2015-5820J

Computing Quality Scores and Uncertainty for Approximate Pattern Matching in Geospatial Semantic Graphs

David J. Stracuzzi
Randy C. Brost
Cynthia A. Phillips
David G. Robinson
Sandia National Laboratories

Alyson G. Wilson
North Carolina State University

Diane M.-K.Woodbridge
Sandia National Laboratories

July 14, 2015

Abstract

Geospatial semantic graphs provide a robust foundation for representing and analyzing remote sensor data. In particular, they support a variety of pattern search operations that capture the spatial and temporal relationships among the objects and events in the data. However, in the presence of large data corpora, even a carefully constructed search query may return a large number of unintended matches. This work considers the problem of calculating a quality score for each match to the query, given that the underlying data are uncertain. We present a preliminary evaluation of three methods for determining both match quality scores and associated uncertainty bounds, illustrated in the context of an example based on overhead imagery data.

Keywords: uncertainty, confidence intervals, statistical models, graphical models, distance metric, image interpretation, graph search

1 Introduction

The number of remote sensing applications has exploded in recent decades. For example, scientists monitor the weather, the state of forest canopies, and changes in habitat ranges for animal species. Likewise governments rely on remote sensing technologies for activities ranging from air and water quality monitoring to assessment of international treaty compliance, such as nuclear non-proliferation. Each of these applications has experienced rapid and consistent increase in both the volume and rate of data collected from remote sensors.

As a side effect of improvements to data collection, the analytical process must also change. Analysts have traditionally relied on manual interpretation of data to identify patterns of interest and to construct models. The explosion in data collection makes the manual approach intractable, so that data analysis must now begin with computational tools. Some computational modeling tools capable of analyzing these data exist. For example, Simonson et al. [15] shows how to automatically co-register multiple images using uncertainty as an indicator of result quality. This combination of a result with a quality assessment is a critical aspect of automated data analysis tools.

More commonly, available tools ignore the role of uncertainty and leave unaddressed its underlying causes. For example, Yue et al. [18] describes a sophisticated effort to analyze complex geospatial patterns in terms of their constituent components, such as a high school in terms of its buildings, grassy areas and paved areas. However, they rely heavily on labeled data, which is typically unavailable, and fail to consider the effects of uncertainty on classification of the components. This is a significant loss to data analysts, as the investigation of uncertainty can provide important information about how well the individual components match the larger pattern. Without it, the data analyst can only guess at the relative quality of individual candidate matches.

This paper reports on initial progress toward the treatment of uncertainty in geospatial semantic graphs [3]. In this context, remote sensing images are processed in terms of primitive semantic objects (image regions), such as buildings, grass, forest, and pavement. These objects are then stored in a graph structure along with the spatial relationships among them. An analyst can then query (search) the resulting graph structure for patterns of interest. Our work considers the problem of calculating a quality score and uncertainty

interval for each query match given uncertain data.

As an example, consider an analyst tasked with scanning wide-area overhead imagery for power plants [3]. Though straightforward for small areas (a single metropolitan area), the task becomes intractable when taken over larger areas (an entire nation), as even a carefully constructed graph search can return many candidate matches of variable quality. Moreover, when we consider finding evidence for construction of new power plants in subsequent data collects, the task becomes even more difficult. In the absence of automatically computed quality and uncertainty information, the analyst must read and interpret the underlying data for each candidate to create their own relative assessments of how well each candidate matches the search pattern. This requires substantial time and effort and reduces the benefits of automatically processing the raw data.

The goal of this work is therefore to provide analysts with information about the relative quality of candidate query matches along with their associated uncertainty intervals. Our work includes two main contributions. The first is a detailed examination of the issues associated with uncertainty analysis in geospatial pattern search applications. Although the preliminary work described here cannot address all of the issues raised, the discussion provides a roadmap for future research. The second contribution is three distinct methods for computing match quality scores with uncertainty intervals. Each method relies on a different set of information and therefore provides a different set of strengths and weaknesses. We demonstrate and evaluate the three methods using an example geospatial search problem that we trace throughout the paper. Finally, we conclude with a discussion of the relative merits of each method, along with a number of directions for future work.

2 Search in Data Using Geospatial Semantic Graphs

Geospatial semantic graphs enable search in remote sensing data by representing both discrete objects with their associated attributes and the relations among them. The graphs support a variety of pattern search operations that capture the spatial and temporal relationships among the objects in the data. Although this paper focuses on imagery, semantic graphs in general can integrate and search a variety of data sources, including multiple imagery types, text, and GPS information.

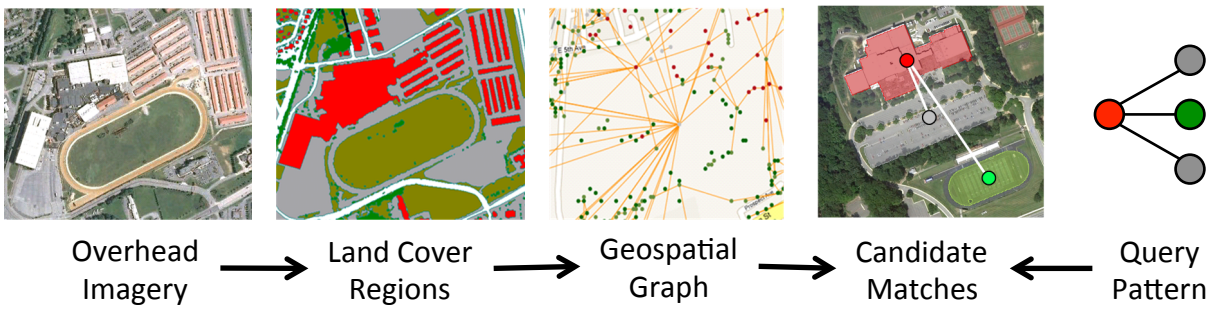

Figure 1: Geospatial semantic graph computation flow.

2.1 Graph Representation, Construction, and Search

Processing begins with a collection of geo-located and orthorectified images, such as from optical, LiDAR, radar, or infrared sources. These are segmented into land cover regions (buildings, trees, water, and so on) which are then classified and labeled (see Kluckner and Bischof [7] and O'Neil-Dunne et al. [12] for examples). The resulting regions form the graph nodes while edges describe relationships among them, such as adjacency or distance. Each node has a rich set of associated attributes describing region properties such as label, area, centroid, and so on. The resulting graph is written to disk to provide a persistent basis for future searches. Figure 1 summarizes the construction and use of the graphs.

Given the constructed graph, an analyst defines a search query as a subgraph template. The nodes and edges of the query graph specify both attribute constraints and topological conditions expected from a proper match. A pair of search algorithms then identifies matches that satisfy both the attribute and topological constraints respectively. See Watson et al. [17] and Brost et al. [3] for a more detailed discussion of search in geospatial semantic graphs. The search algorithms produce an unordered set of subgraph matches, such that each match satisfies the query template.

These matches are then displayed to the analyst, cueing them to review specific areas of interest. To reduce the number of missed areas of interest, the analyst may specify a query with wide parameter ranges, and designate some components as optional. This reduces the number of false negatives, but also increases the number of false positives and the variability in match quality. The motivation for the work presented here is that without some sort of quality scoring or rank ordering on the search results, reviewing the large numbers of

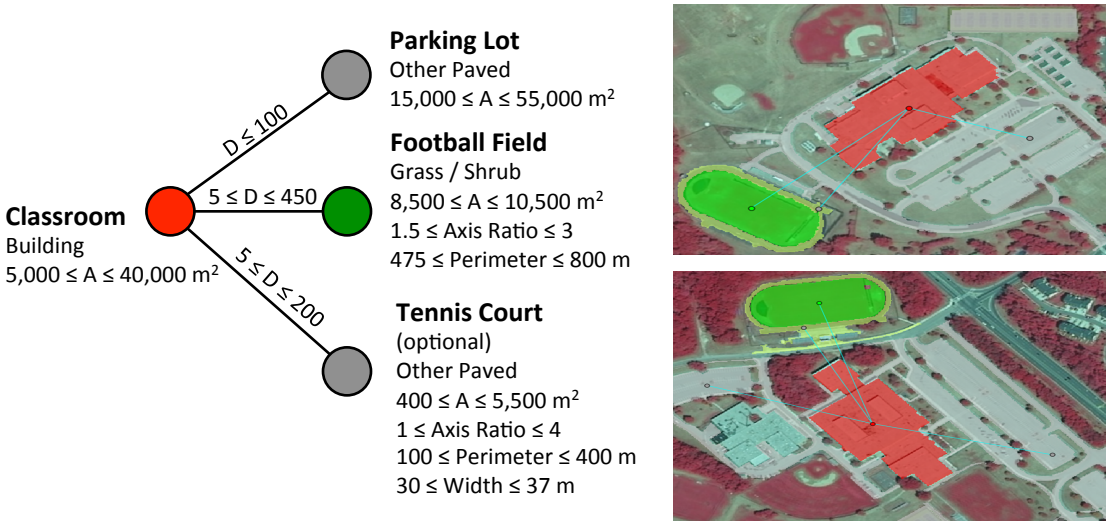

Figure 2: The high school query template (a) and two candidate matches (b).

matches becomes impossible. The following section presents an example to help clarify the preceding description.

2.2 A Simple Example: The High School Search

In the remainder of the paper, we discuss an example search for high schools in Anne Arundel County, MD, an area of 1,523 km^2. Following the procedure summarized above, we begin with a 0.6 meter resolution land cover image generated from overhead optical imagery, LiDAR, and GIS road network data using methods described by O'Neil-Dunne et al. [12]. The land cover region labels are building, trees, grass/shrub, dirt, water, road, and other paved. From this, we constructed a graph with over 1.2 million nodes.

Next, we defined the high school query template shown in Figure 2 (a) as a building node corresponding to the classroom building, with an associated paved node for the parking lot and grass node for the football field. The node attribute constraints correspond to the land cover labels and region area, eccentricity, perimeter, and width as shown in the figure. The template also includes optional tennis courts, meaning that failure to match them does not disqualify a candidate subgraph. Topological constraints include the links among the nodes with associated distance attributes.

Given this query, our search algorithm returned 40 high school match candidates. Fig-

ure 2 (b) shows two example matches. Notice that the layout and shape of the buildings and parking lots are quite different; this flexibility is a key benefit of the semantic graph approach. For the purposes of this paper we defined a loose query, so the returned matches included a number of false positives.

The original query reported in Watson et al. [17] correctly found the 12 public high schools, with only two false positives. As a test case for quality scoring and uncertainty analysis, such an accurate query design is undesirable for two reasons. First, tuning the query parameters required numerous iterations. However, quality scores and uncertainty are most informative during initial stages of this tuning process because they help the user to quickly identify true positives and narrow the scope of the search. Second, many queries may never produce such accurate results, so performance in the context of ambiguous results is critical. For example, a query designed to identify big-box retailers may not be able to separate them from supermarkets, mega-churches, and furniture warehouses.

2.3 Sources of Variation and Uncertainty in Match Quality

Several factors may influence the level of uncertainty in match quality evaluations. For example, the raw data captured by the physical sensors vary due to environmental conditions during collection, resulting in noisy data. Likewise, integration of multimodal data sources, which often have different sampling rates, also contributes to noise and uncertainty (see Nguyen [10] for a review of issues and methods). Together, these contribute to uncertainty in the results produced by segmentation and classification algorithms. As a result, the node and edge attributes in the semantic graph are all uncertain.

Variance in the concept described by the query pattern also contributes to uncertainty. For example, the set of public high schools in Anne Arundel County exhibit wide variation in the attributes used to describe them. The concept of *high school* as represented in the semantic graph is therefore not unique; it admits objects such as parks and sports facilities. This is partly a side effect of limitations on graph content and on the graph query language. However, it also stems from the variability associated with the concept of high schools in general. As a result, the match quality uncertainty of all candidate high schools increases.

A final source of variability results from candidate topology. For example, while a match

to the template in Figure 2 containing a classroom building, parking lot, and football field remains valid, another match that includes these plus a tennis court has stronger evidence. Similarly, the number of replications of an element also influences match quality. A candidate high school with one football field is a valid match, while a candidate with separate practice and game fields may be stronger. Yet another candidate with 10 football fields might be more indicative of a recreational complex than a high school. We do not consider topological issues further in this paper.

Much of the discussion so far has focused on the relationship between uncertainty and false positive match candidates. An equally important question concerns whether the presence of uncertainty increases false negatives. In general, false negatives due to concept variance are always a risk. For example, a rural high school with relatively few students may not be recognized by the search algorithm due to an unusually small building and a football field that is indistinguishable from other fields. The presence of attribute uncertainty is partially mitigated by adjusting the search algorithm to accept any candidate whose attribute uncertainty intervals overlap the query pattern. This may increase the number of candidates, but avoids false negatives to the extent possible.

2.4 Match Quality Scoring Desiderata

To summarize, we are given an attributed graph with uncertain attribute values from which we must find instances of a specified query. The goal is to compute, for each match candidate, a quality score and associated uncertainty interval that indicates the degree to which it matches the given template. The quality scores prioritize candidate matches for further consideration, while the uncertainty estimates indicate the strength of evidence supporting the score. In the remainder of this section, we discuss several issues that the scoring and uncertainty methods must address in the context of geospatial semantic graphs.

Domain knowledge plays a critical role in distinguishing between high- and low-quality matches. Quality and uncertainty methods should therefore incorporate whatever information is available, including the query specification, labeled training data, and expert knowledge. Labeled examples and elicited knowledge in particular can provide valuable information about concept variability. As a corollary, any required expert knowledge elic-

itation and data labeling must also be practical. For example, domain experts are notoriously bad at marginalizing over distributions, and image analysts cannot be expected to label hundreds of examples for each query specification. Evaluation methods that require eliciting such knowledge or examples are therefore undesirable.

User interpretation of the quality scores and uncertainty intervals also impacts the selection of computation methods. Geospatial semantic graph users are intended to be analysts with expertise in the sensor domain, as opposed to computer scientists and statisticians. Provided scores and intervals should therefore make intuitive sense. For example, scores should provide a smooth and monotonic response to changes in underlying components.

Finally, the generality of the semantic graph approach across data sources implies that the quality scoring methods should also generalize well. For example, we prefer to avoid tunable modeling parameters (excluding parameters that describe the search pattern) that require empirical adjustment from application to application. Similarly, assumptions made by the quality and interval estimation methods should be theoretically justifiable to the extent possible. The intent here is to ensure our methods have some expectation of working in most cases, and that we can qualify cases in which we do not expect them to work.

3 Methods for Quality and Uncertainty Estimation

Having identified a set of desirable solution properties, we now describe three approaches to quality scoring and uncertainty estimation. Each method relies on a subset of the uncertainty and domain knowledge sources discussed in the preceding section. Elicitation-based statistical models address rare query patterns by assuming a modeling distribution and then populating its parameters via expert knowledge, optionally augmented with labeled training examples. Bayesian graphical models address queries for which labeled examples are readily available by modeling the joint probability distribution. Finally, distance-based quality metrics determine the degree of match between the candidate subgraph and query template. In the following, we explore each method in greater detail, highlighting relationships to the uncertainty sources and solution properties outlined in the previous section.

3.1 Elicitation-Based Statistical Models

One approach to evaluating search results is to calculate the probability that the candidate represents an instance of the query pattern. Toward this end, we select an appropriate distribution and then elicit its parameters from a domain expert. The use of elicited knowledge has two advantages. First, it supports evaluation of rare patterns that lack training examples. Second, it drives the evaluation to consider the analyst's desired concept (such as *high school*) as opposed to the requested query pattern (a building adjacent to a paved area and so on). The difference is subtle, but important in the context of concept variability. The elicitation-based method described below could also be adapted to incorporate uncertainty in the attribute values. We have not done so here because the land cover maps used to create the graph do not contain the required boundary and label uncertainty. Adapting O'Neil-Dunne et al's [12] segmentation and classification methods to produce the required uncertainties remains a point of future work.

Our approach follows methodology proposed by Bedrick et al. [1]. Denote the feature vector describing the i^{th} example match by $\mathbf{x}_i = \{x_1, \ldots, x_n\}$. Note that the features need not be equivalent to the list of attributes that describe the matched nodes. For example, it may exclude some attributes, and include interaction terms. The goal is to calculate the probability $P(\mathbf{x}_i)$ that candidate $\mathbf{x}_i$ is an instance of the query pattern, and to quantify our uncertainty about $P(\mathbf{x}_i)$. Given that we are modeling the probability of a binary outcome, high school or not, we assume that the following logistic regression structure relates our probability $P(\mathbf{x}_i)$ to the feature vector $\mathbf{x}_i$:

$$\log\left(\frac{P(\mathbf{x}_i)}{1 - P(\mathbf{x}_i)}\right) = \sum_{j=0}^{n} x_{ij}\alpha_j \tag{1}$$

where j indexes the features, α_0 corresponds to an intercept term with $x_{i0} = 1$, and $\alpha_1, \ldots, \alpha_n$ correspond to main effects and/or interaction terms between features. Solving Equation 1 for $P(\mathbf{x}_i)$ yields

$$P(\mathbf{x}_i) = \frac{\exp(\sum_{j=0}^{n} x_{ij}\alpha_j)}{1 + \exp(\sum_{j=0}^{n} x_{ij}\alpha_j)}. \tag{2}$$

Given these equations, we next elicit values for $n + 1$ probabilities $P(\mathbf{x}_i)$ corresponding to selected feature vectors $\mathbf{x}_i$. We then use these elicited distributions to induce a

joint probability distribution for the αs, and use Monte Carlo simulation to determine a probability for any new feature vector $\mathbf{y}$.

To elicit the required information, we use Latin Hypercube Sampling as implemented using the `maximinLHS` function in R [4, 16] to select n well-separated feature vectors, $\mathbf{x}_i, i = 0 \ldots n-1$. To this, we added one feature vector representing the mean of all features from the specified ranges, for a total of $n+1$ vectors. We then create an abstract image for each of the $n+1$ feature vectors plus a reference vector with value equal to the mean of the preferred feature ranges as specified in the search template.

For the high school example, the images included rectangles with sides in a golden mean ratio scaled to the desired area for classroom buildings and parking lots, which have only area features. Similarly, all Ann Arundel County high schools have a running track around the football field, so we represented the football field as the interior of a elliptical track. Although running tracks are ovals, using ellipses simplified the calculation of the required long axes given an area and an axis ratio (both of which are included in the feature vector). We used a consistent scale for all distances and areas, adjusting the layout to keep distances consistent and to fit on a single powerpoint slide. We also placed a rectangle consistent with a car in the parking lot for added perspective. Finally, we asked an expert to provide a pair of values for each layout: the median and either a "surprisingly" high or low probability that the layout corresponds to a high school.

Next, these probability values were set to the median and the high or low values were set to the 0.9 or 0.1 percentile respectively of the beta distributions for each $P(\mathbf{x}_i)$. Having elicited for each of the $P(\mathbf{x}_i)$, we assume that their beta prior distributions are independent (see Bedrick et al. [1] for extensive discussion on this point). Let $\mathbf{X}$ be a $(n+1) \times (n+1)$ matrix that has as its rows the feature vectors from the elicitation. We have

$$\begin{pmatrix} \log(\frac{P(\mathbf{x}_0)}{1-P(\mathbf{x}_0)}) \\ \log(\frac{P(\mathbf{x}_1)}{1-P(\mathbf{x}_1)}) \\ \vdots \\ \log(\frac{P(\mathbf{x}_n)}{1-P(\mathbf{x}_n)}) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} \boldsymbol{\alpha}$$

and inverting, we have

$$\boldsymbol{\alpha} = \mathbf{X}^{-1} \begin{pmatrix} \log(\frac{P(\mathbf{x}_0)}{1-P(\mathbf{x}_0)}) \\ \log(\frac{P(\mathbf{x}_1)}{1-P(\mathbf{x}_1)}) \\ \vdots \\ \log(\frac{P(\mathbf{x}_n)}{1-P(\mathbf{x}_n)}) \end{pmatrix}. \tag{3}$$

We can now use Equation 3 to generate random draws from the distribution of $\boldsymbol{\alpha}$ by making random draws from each beta distribution describing $P(\mathbf{x}_i)$. To do so, first generate R random draws from each beta distribution describing $P(\mathbf{x}_i)$. Denote these as $P(\mathbf{x}_i)^{(r)}, r = 1, \ldots R$. Next, substitute these values into Equation 3 to get random draws $\boldsymbol{\alpha}^{(r)}$. For any feature vector $\mathbf{v}$, we can obtain random draws from the distribution $P(\mathbf{v})$ by

$$P(\mathbf{v})^{(r)} = \frac{\exp(\sum_{j=0}^{n} v_j \alpha_j^{(r)})}{1 + \exp(\sum_{j=0}^{n} v_j \alpha_j^{(r)})}. \tag{4}$$

The above procedure yields a distribution for $P(\mathbf{v})$ given the expert elicitation, from which we can calculate a variety of statistics to support decision making.

We can also incorporate labeled examples using Bayes' Theorem by treating each example as an independent Bernoulli observation. Suppose that we had two labeled examples: $\mathbf{y}_1$ is labeled as a high school, and $\mathbf{y}_2$ is not a high school. Using Bayes' Theorem, our posterior distribution for $P(\mathbf{x}_0), \ldots, P(\mathbf{x}_n)$ becomes

$$f(P(\mathbf{x}_0), \ldots, P(\mathbf{x}_n) \mid \text{data}) \propto \text{logit}^{-1} \left(\mathbf{X}^{-1} \begin{pmatrix} \log(\frac{P(\mathbf{x}_0)}{1-P(\mathbf{x}_0)}) \\ \log(\frac{P(\mathbf{x}_1)}{1-P(\mathbf{x}_1)}) \\ \vdots \\ \log(\frac{P(\mathbf{x}_n)}{1-P(\mathbf{x}_n)}) \end{pmatrix} \mathbf{y}_1 \right)$$
$$\left(1 - \text{logit}^{-1} \mathbf{X}^{-1} \begin{pmatrix} \log(\frac{P(\mathbf{x}_0)}{1-P(\mathbf{x}_0)}) \\ \log(\frac{P(\mathbf{x}_1)}{1-P(\mathbf{x}_1)}) \\ \vdots \\ \log(\frac{P(\mathbf{x}_n)}{1-P(\mathbf{x}_n)}) \end{pmatrix} \mathbf{y}_2 \right) \left(\prod_{i=0}^{n} P(\mathbf{x}_i)^{\gamma_i - 1} (1 - P(\mathbf{x}_i))^{\delta_i - 1} \right)$$

where γ_i and δ_i correspond to the parameters for $P(\mathbf{x}_i)$'s beta distribution. Now to get samples from the posterior distribution of $\boldsymbol{\alpha}$, we first draw samples, $P(\mathbf{x}_i)^{(r)}$, from the above posterior distribution for $P(\mathbf{x}_0), \ldots, P(\mathbf{x}_n)$ using a Markov chain Monte Carlo algorithm.

Then we use the Monte Carlo algorithm described above with Equation 3 to get samples from the posterior distribution of $\boldsymbol{\alpha}$, and Equation 4 to get samples for the posterior of some new example $P(\mathbf{y})$.

3.2 Graphical Models

Like the elicited model described above, graphical models calculate the probability that a candidate subgraph, described by a feature vector, represents an instance of the query pattern. While the elicited approach uses a set of beta distributions to induce a joint probability distribution for the regression parameters $\boldsymbol{\alpha}$, graphical models use conditional independence to factor the joint over the features. Unlike the elicited approach, graphical models depend only on labeled examples, which simplifies model parameterization. The labeling process may also be more reliable than the elicitation described above. Nevertheless, both methods evaluate candidates with respect to the desired concept versus the specified query and both estimate uncertainty due to conceptual variability.

A variety of graphical modeling approaches are available, including directed Bayesian models and undirected Markov models [8, 13]. The primary considerations in choosing a model are the dependencies that need to be represented and the amount of training data available. In this work, we assume that only tens of examples may be available and that dependencies among variables are not well known. Given these constraints, we elected to use the naïve Bayes model which assumes that all variables (attribute values) are conditionally independent given the class variable (in this case, high school or not). Naïve Bayes requires relatively few training examples, and model structure is trivially specified.

Although the independence assumption will clearly be violated, naïve Bayes often produces useful results. Specifically, the impact of violating the independence assumption can be viewed as double-counting evidence. This causes the output probabilities to skew toward the most likely class, but does not generally change the predicted class. In the context of assessing the relative quality of a set of candidate matches, this suggests that naïve Bayes will still push the true positives toward the top, although the ordering may be altered. Domingos and Pazzani [5] provide a detailed study of the properties of naïve Bayes under independence assumption violations.

For a candidate feature vector $\mathbf{x}$, naïve Bayes calculates

$$P(c|\mathbf{x}) = \frac{P(c)\prod_{j=0}^{n-1} P(x_j|c)}{P(c)\prod_{j=0}^{n-1} P(x_j|c) + P(\neg c)\prod_{j=0}^{n-1} P(x_j|\neg c)}, \tag{5}$$

where c is the class value (high school or not), $P(c)$ is the prior probability of observing an instance of c, $P(x_j|c)$ is the conditional probability of observing feature x_j given that $\mathbf{x}$ is an instance of c, and $\neg$ represents logical negation.

The $P(c)$ and $P(x_j|c)$ probabilities are estimated from observed frequencies in the t supervised training examples $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. This can be done by treating the features as continuous variables (see John and Langley [6], for example) or by discretizing them into categorical variables and simply counting. We chose the latter, which is more appropriate for small numbers of examples. The simplest approach is to use the attribute constraint values from the query template as discretization cut-offs ($x_j < \min_j$, $\min_j \leq x_j \leq \max_j$, $\max_j < x_j$) plus a special value when the node is missing.

We also apply Bayesian m-estimates to avoid degenerate cases in which some conditional probability estimate equals zero. The basic estimate, $P(x_j|c) = \frac{n_j}{n_c}$ where n_j is the number of class-c examples from $\mathbf{Y}$ that have feature x_j, and n_c is the number of examples from class c, fails when $n_j = 0$ (no observations in the data). The m-estimate ensures that none of the probabilities is exactly zero by calculating $P(x_j|c) = \frac{n_j + mp}{n+m}$ where p is a prior and m determines how heavily to weight p relative to the observed data (equivalent sample size). We chose $m = 1$ and $p = \frac{1}{k}$ where k is the number of discrete values of x_j. See Mitchell [9] for additional details on naïve Bayes and m-estimates.

Equation 5 provides a point estimate for $P(c|\mathbf{x})$. We derive a distribution for these probabilities by applying Monte Carlo sampling over the set of training examples. Specifically, we repeatedly subsample the training examples $\mathbf{Y}$ and compute $P(c|\mathbf{x})$ over each sample. For each sampled training set, we balance the number of positive and negative examples such that the prior $P(c)$ remains approximately constant across all subsamples. As with the elicited statistical model, the resulting distribution provides information about the uncertainty in the probability estimate due to the training data. As an alternative, Pronk et al. [14] show how to calculate the uncertainty without subsampling the training data. However, their approach makes a second use of the independence assumption whose

effect is more difficult to understand than above. An empirical comparison of the two approaches remains a point of future work.

3.3 Distance-Based Quality Metrics

While the preceding methods focused on calculating the probability that a given match represents an instance of the query pattern, the distance-based approach calculates a score indicating how well a candidate matches the search template. Note the important differences. The probabilistic methods attempt to model the underlying notion of the target query using labeled examples and elicitation. To some extent, they can account for aspects of the query not encoded into the search template. Conversely, distance-based methods measure only the similarity of the candidate subgraph to the template. As a result, the approach can map any subgraph to a quality score and requires no labelled examples and minimal elicitation, but its accuracy is limited by the accuracy of the template.

We base a subgraph's quality score on the distance between the candidate and the query template. To do so, we first extend the notion of template attribute constraints to include both *preferred* values, which have zero distance, and *allowable* values, whose distance increases monotonically as the attribute value moves away from the preferred range. (The search algorithm returns all subgraphs whose attribute values fall within the allowable range.) We then construct a distance score by averaging over the attribute distances for each candidate's nodes and edges. Finally, we convert the distance to a quality score by inverting (low distance equals high quality) and scaling into the range zero to one.

Figure 3 shows the general schema for the distance functions. Given a value x_{ijk} for attribute k of node (or edge) j from candidate match i, the function computes the required distance, d_{ijk}. Attribute values outside of the allowable range are assigned a large value $d_{j,missing}$, since the corresponding node or edge is considered absent. The rate at which distance increases as attribute values deviate from the preferred range is controlled by the parameters $d_{jk,allow,min}$ and $d_{jk,allow,max}$, whose selection we discuss below. These parameters provide analysts with nuanced control in specifying the distance function's shape.

Given the distances associated with each attribute specified in the query, we then calculate the total distance D_i of candidate i as the average of these per-attribute distances.

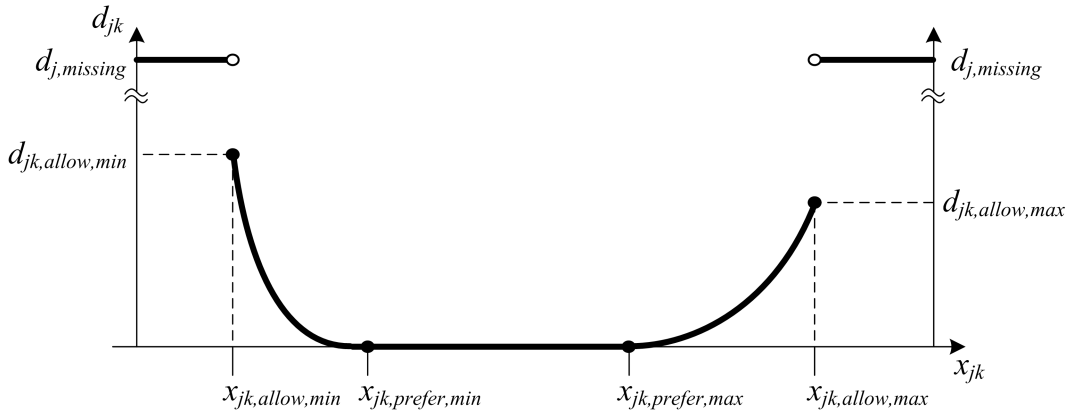

Figure 3: Distance function and associated parameters for an attribute k of node j.

However, we desire a function where quality scores collapse if a required element is missing (such as a high school building). To accomplish this, we define a per-element distance for node or edge j of match i as

$$d_{ij} = \begin{cases} \sum_{k=1}^{n_j} d_{ijk} & \text{If all attributes are within allowable range.} \\ d_{j,missing} & \text{If any attribute is outside allowable range, or if node } j \text{ is missing completely.} \end{cases} \quad (6)$$

where n_j is the number of attributes of node (edge) j, d_{ijk} is determined by the distance function, and $d_{j,missing}$ is large enough to dominate D_i. Importantly, $d_{j,missing}$ must be carefully chosen when element j is optional in the query. We discuss how to select this value below.

Equation 6 yields a total distance for candidate i of

$$D_i = \frac{\sum_{j=1}^{m} d_{ij}}{\sum_{j=1}^{m} n_j},$$

where m is the number of elements (nodes and edges) in the query template and n_j is the number of attributes for node or edge j. We then scale the total distance into an overall quality score between 0 and 1 with $q_i = \frac{1}{1+D_i}$. If all template elements are present and all attribute values are within their preferred ranges, then $D_i = 0$ and $q_i = 1$. As some parameters fall outside the preferred range, D_i increases and q_i decreases. When an attribute of any required element falls outside its allowable limits, then D_i becomes large and drives q_i to near zero.

In the context of a search problem, the analyst defines a distance function $d_{jk}(x_{jk})$ for each attribute k of each template element j. She then selects desired quality scores $q_{jk,allow,min}$ and $q_{jk,allow,max}$, corresponding to the score desired when the attribute value is at the allowable limit, but the match is otherwise perfect. For example, if a match i is perfect (zero distance) except that attribute k of element j is at the minimum limit, then

$$q_i = q_{jk,allow,min} = \frac{1}{1 + \frac{d_{jk,allow,min}}{\sum_{j=1}^{m} n_j}}.$$

We then solve for $d_{jk,allow,min}$ given the selected $q_{jk,allow,min}$. We obtain $d_{jk,allow,max}$ similarly.

To provide a smooth falloff in quality, the portion of the function $d_{jk}(x_{jk})$ connecting the preferred range to the allowable limit is constrained to have zero slope at the preferred range endpoints. The user can optionally control the shape of the falloff by selecting intermediate control points; we omit these details here.

The user completes the function $d_{jk}(x_{jk})$ by selecting the value $d_{j,missing}$. If element j is required, then they choose $d_{j,missing}$ to be large enough to overwhelm the contributions of all attributes within their acceptable limits. If the element j is optional, then the user selects a quality, $q_{j,missing}$, desired if the match is missing element j but is otherwise perfect. Given this value, we compute $d_{j,missing}$ analogously to $d_{jk,allow,min}$. Note that $d_{jk,allow,min}$ and $d_{jk,allow,max}$ must be strictly less than $d_{j,missing}$, to prevent a reversal of quality scores.

The above formulas compute a quality score. We estimate the uncertainty in the score by using Monte Carlo simulation over the uncertainty in the feature values to derive a score distribution. For simplicity, we restrict ourselves in this paper to the uncertainty in observed image feature shape. We model shape variation as an uncertainty in the location of the geometric boundary of the feature. Ideally, the underlying land cover map would provide a probability distribution over boundary locations. Since this is not available in our data, we assume that the boundary location is normally distributed with standard deviation σ_e, reflecting both sensor and pre-processing effects. We then sample boundary locations from within $\pm 3\sigma_e$ of the nominal boundary location, and use these to calculate attributes such as distance, area, perimeter, axis ratio, and width.

We apply simple approximations to generate samples, inspired by the intuition that the true shape is within a band of $\pm 3\sigma_e$ defined along the perimeter of the observed shape. For example, if a match has two nodes separated by a minimum distance d_{min}, then in

our Monte Carlo simulation we sample the corresponding attribute from $d_{min} \pm 6\sigma_e$, since shape error occurs for both nodes, doubling the variation in the inter-node distance. Other features such as perimeter and areas are defined analogously. These simple models could clearly be improved, but they have the advantages that they are responsive to the actual object shape and the percentage error varies appropriately with feature size.

The uncertainty calculations described above quantify the impact of data collection and processing errors on the quality scores. Importantly, distance metrics can be extended to incorporate variation in the target pattern, as illustrated by Berger-Wolf et al. [2]. Finally, note that the geospatial graph search algorithm described by Brost et al. [3] allows matches with multiple copies of both required and optional nodes. We do not address this complication here.

4 Results on the High School Problem

We tested all three methods described in Section 3 on the high school problem. To generate the set of match candidates, we applied the template shown in Figure 2 to the Anne Arundel County graph, which returned 40 candidate matches. Table 1 shows a subset of these results. Note that for some candidates, the nodes matched for football fields and tennis courts are false positives, and some candidates do not match the tennis court at all.

Our match evaluation methods all rely on the allowable and preferred value ranges in

	Class	Football Field			Parking	Tennis	Distance		
Identifier	Area	Area	Ratio	Per	Area	Courts	CB-FF	CB-PL	CB-TC
High School 1	16136	10235	2.23	562	46364	···	75	0	23
High School 2	14802	9530	2.44	584	32196		95	0	26
High School 3	13640	8957	2.36	773	41296		46	0	77
Museum	7522	9904	2.30	499	23798		29	29	170
Training Center 2	11411	9904	2.30	499	16301		26	26	69
Middle School 1	9402	9530	2.44	584	32196		314	18	-1
···									

Table 1: A subset of the match candidates returned for the high school query. Identifiers in boldface indicate true positives, while -1 indicates a missing value. CB is Class Building, FF is Football Field, PL is Parking Lot, and TC is Tennis Court.

Node	Attribute	Min Allowable	Min Preferred	Max Preferred	Max Allowable
Classroom	Area	5000	12189	21363	40000
Football Field	Area	8500	8963	10070	10500
	Axis Ratio	1.50	2.17	2.47	3.00
	Perimeter	475	507	611	800
Parking	Area	15000	23280	49303	55000
Tennis Courts	Area	400	1168	4390	5500
	Axis Ratio	1.00	1.05	3.30	4.00
	Perimeter	100	154	354	400
	Width	30.0	32.2	36.5	37.0
Distances	Class-FF	5	48	254	450
	Class-Park	0	0	30	100
	Class-TC	5	26	77	200

Table 2: Preferred and allowable values for high school node attributes.

evaluating match candidates. In practice, these values are selected by the analyst when constructing the search template. Table 2 lists the values used throughout the remainder of the paper. Note that the allowable range corresponds to the required values used by the search algorithms when identifying candidate matches.

In the following, we evaluate and provide uncertainty intervals for each match candidate, using the evaluation results to rank the candidates. Ideally, the true high schools should score near the top of the list and guide image analysts to minimize the number of non-high schools considered. The task is similar to the page ranking that web search engines perform, with the critical difference that we lack a query-independent evaluation criterion such as page linkage.

4.1 Elicited Beta Distribution Model

The elicited model requires one elicitation (of two values) per feature. To reduce the number of elicitations, we excluded some of the features defined by the query template. The final feature set includes classroom area, parking lot area, football field area, distance from football field to building, distance from parking lot to building, football field axis ratio, presence of tennis courts, and given the courts, the tennis court area, tennis court width, and distance from tennis courts to the building. We also included an interaction term between building area and parking lot area for a total of 12 required elicitations (including

the constant term). The latin hypercube uses both the preferred and allowable boundary values for each node attribute when selecting elicitation points.

We performed two simulations: one with the expert-elicited distribution, and one with independent uniform prior distributions assigned to each $P(\mathbf{x}_i)$. For each of the 40 matches returned by the search, we randomly selected 10 training sets of size 6, 12, 18, 24, 30, and 36 from the remaining 39 matches. We selected positive and negative examples in a ratio of 1:2 to agree with the full set of 40 matches. The probability distributions were then calculated based on $R = 40,000$ MCMC draws.

The top panel of Figure 4 summarizes the results based on 36 training examples and the expert elicited distribution. The plot shows the 5^{th}, 50^{th}, and 95^{th} percentiles for all 40 candidate matches as calculated from the training set that yielded the distribution with the 5th largest median (of the 10 training sets). The 40 matches are sorted along the x-axis by the median values, yielding an ordering that an analyst would see. Of the 13 high schools, 12 sorted into the top 15 results, with High School 13 as the only exception. Several middle schools also sorted near the top. This is expected, as some of the middle schools are sized similarly to, co-located with, and share athletic facilities with a high school. The rank ordering remains unstable until approximately 24 training examples (not shown), though most high schools tend to rank in the top 25.

Figure 5 illustrates the relationship between the calculated conditional probability, uncertainty intervals, and number of training examples using learning curves for both the elicited and uninformative prior models on three selected search results. For smaller amounts of training data, including the elicitation-only model (zero training examples), the spread between the 5^{th} and 95^{th} percentiles is predictably larger. Finally, our results showed that the elicited priors improved the candidate ranking relative to the uniform prior when training examples were sparse, but did not consistently reduce the width of the uncertainty intervals. Larger numbers of examples (≥ 30) tended to dominate the effects of the priors.

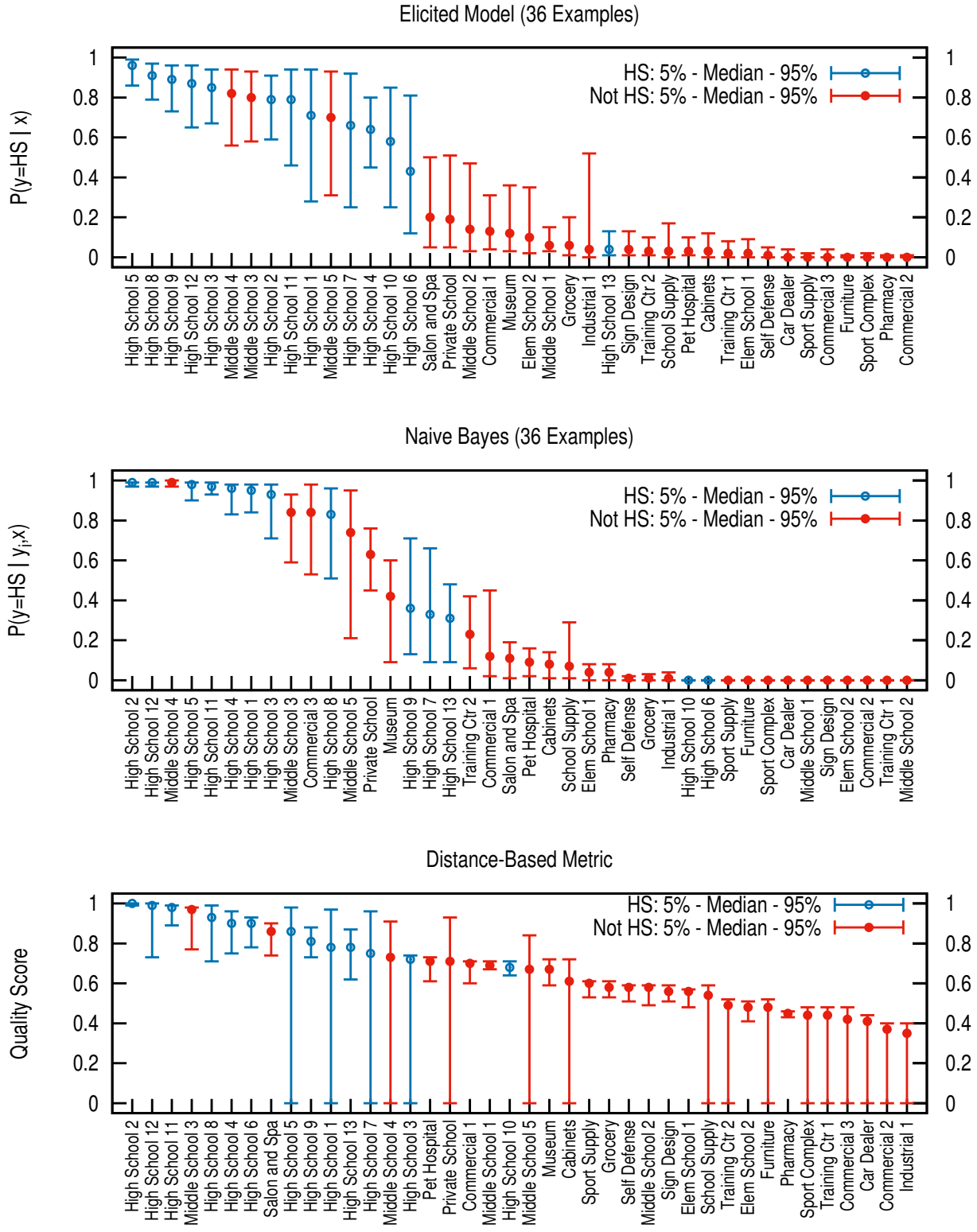

Figure 4: Results for the three methods over all 40 high school candidates.

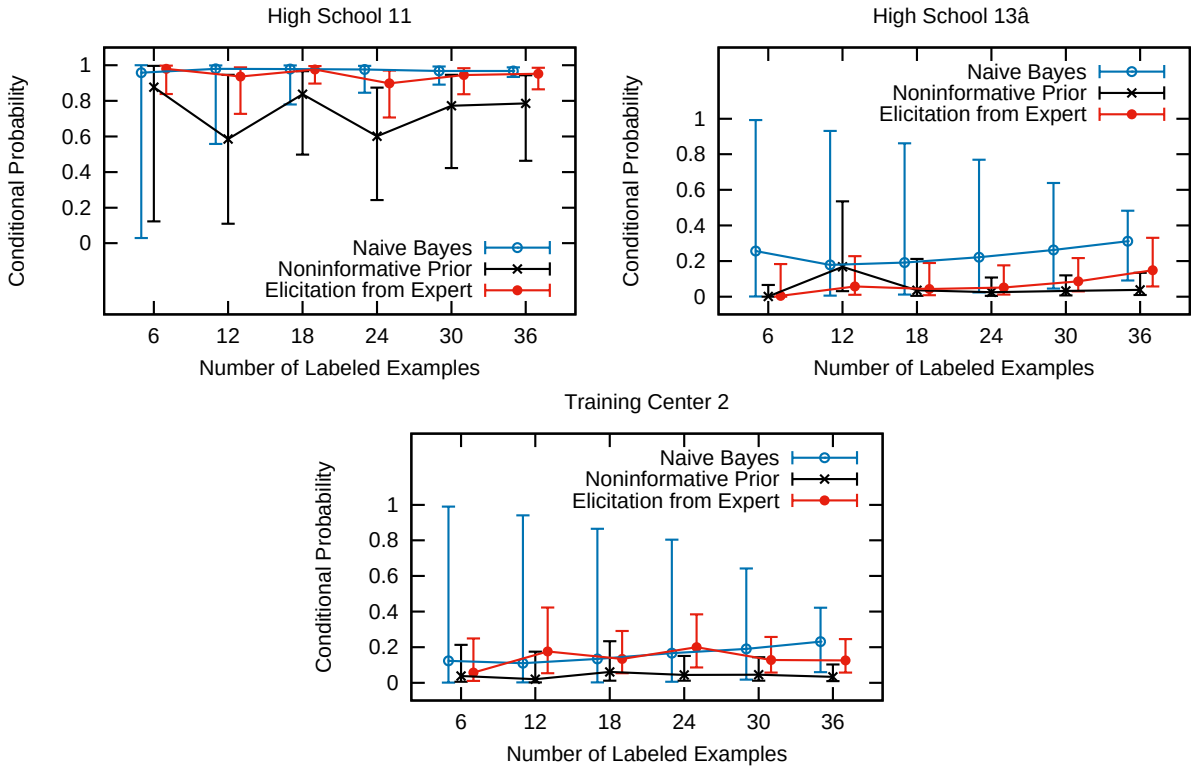

Figure 5: Learning curves for the elicited prior, uninformative prior, and naïve Bayes models for three selected examples. Points represent median values and the bars span the 5th percentile to the 95th percentile. Points for the three methods are offset slightly in the plot to increase readability.

4.2 Naive Bayesian Model

We applied naïve Bayes using all 12 numeric attributes included in Table 2 after discretizing according to the indicated boundary values. We then used the Monte Carlo methods described in Section 3.2 to construct distributions for the probability that a match represents a high school given the observation and training sets of size 6, 12, 18, 24, 30, and 36. From these, we collected statistics similar to those collected for the elicited model.

The center panel of Figure 4 shows results for 36 training examples. All high schools are ranked highly except two, with 11 of the 13 high schools in the top 17. Middle schools constitute most of the highly ranked non-high schools. Notice the steep drop in median probability values between the 10^{th} and 15^{th} results. This is a side-effect of independence assumption violations, which pushes most of the probability mass toward the extremes. That

the drop is not more precipitous suggests that the attributes are reasonably independent. Likewise, the bulk of the uncertainty resides with a handful of the candidates. Specifically, the ten candidates that have median probability estimates between 0.15 and 0.85 also have the highest uncertainty. This may also be a side effect of the independence assumption: as violations push probability estimates toward the extremes, they also increase variability due to minor differences in training data.

The learning curves in Figure 5 show uncertainty predictably falling while median probabilities remain stable as the number of training examples increases. There are some exceptions to this trend, however. Also noteworthy is that naïve Bayes tends to have larger uncertainty intervals than either of the elicited models given few training examples. However, the intervals become comparable for larger amounts of training data.

4.3 Distance-Based Quality Metrics

The following results are based on the assumption that $3\sigma = 0.3$ meters (see Section 3.3), which corresponds to one half pixel width. We assume that image segmentation makes the correct decision nearly all of the time (pixel classification accuracy was $> 97\%$). A geospatial boundary occurs when two pixel types (such as pavement and dirt) are adjacent in the land cover map. The true boundary could be up to one half pixel in either direction.

The lower panel of Figure 4 shows the results for the distance-based approach. The 13 high schools all appear in the top 20 results, though middle schools represent an important source of confusion. Importantly, the median quality scores do not drop as steeply as the probabilities. This is an advantage in that it indicates just how similar the true positives are to the false positives, but also a disadvantage in that the numeric scores do not provide a clear indication that some matches are better than others.

The uncertainty intervals require detailed consideration. Many of the 5^{th} percentile scores are near zero, even for candidates with high medians. The problem arises because the search returns results where some parameters are near the allowable limits. When the Monte Carlo simulation dithers these by $\pm 3\sigma$, some of the generated samples fall outside the allowable limits, driving the quality scores to near zero. For poor matches, it's unsurprising that the dithering would cause the quality score to fall. However, when the uncertainty

bars on high-scoring matches extend to zero the user may perceive an error in the results. In practice, large uncertainty on high-scoring matching suggests that the allowable limit is too close to the preferred range. The dithering process takes an attribute within or near to the preferred range and samples it outside the allowable limit.

The sharp drop in 5^{th} percentile scores follows from the discontinuity in the distance functions. When attributes such as area and perimeter are near the allowable cut-offs, sampling over a distribution of possible boundary locations causes some of the samples to fall outside of the allowable range, giving them a quality of zero. One possible solution is to ensure that the allowable limits are more than 3σ from the preferred range. We can also improve the sampling windows, which may be unreasonably large for some attributes. For example, perimeter values may vary excessively given the simple $\pm 3\sigma$ estimation, requiring a more sophisticated calculation.

5 Discussion

One salient feature of the graphs in Figure 4 is the disagreement in rank-ordering among the three scoring methods. Several factors contribute to these differences. First, note that the notion of a well-defined "correct" ranking does not exist for the high school search task. Clearly true high schools should rank highly while other locations receive lower rank, but the desired ordering among true high schools is unclear. Likewise, it's unclear how a sports complex should compare to a furniture store that happens to be located adjacent to a park.

With this in mind, compare the top ten results from each scoring method. The elicited and naïve Bayes models agree on eight. Naïve Bayes includes one commercial building that the elicited model does not, but the other disagreement simply swaps one high school for another. Similarly, the elicited and distance models agree on seven of ten, with two of the remaining three being high school swaps. A comparable pattern holds for naïve Bayes and the distance metric. From a practical point of view, the three methods produce very similar results. A related argument can be made for low-ranking candidates with the added consideration that in most cases, the median probabilities and quality scores differ by insignificant amounts, which implies that the specific ordering is arbitrary.

Some inconsistencies between the probabilistic models stem from a lack of data. Given

the number of attributes and threshold values (see Table 2), 36 training examples cannot cover all possible combinations. As a result, portions of the models may rely entirely on the prior, giving the elicited model an advantage over naïve Bayes.

Computationally, the elicited model requires significantly more cycles than the other methods due to its reliance on MCMC algorithms. It also requires operator control to ensure proper MCMC burn-in and to conduct the knowledge elicitation. Creating the elicitation examples required several hours, though it may be possible to automate the process. These two issues imply that the elicited approach may not be appropriate if the analyst needs immediate results. Naïve Bayes requires only that the analyst label examples, and can easily scale computationally to large numbers of candidate. The distance-based metric will also scale up easily.

A clear next step of our work is to extend the probabilistic methods to handle uncertainty in geospatial boundary location, and all three methods need extension to label uncertainty. In conjunction, the land cover map must be improved to provide uncertainty information calculated at the pixel level for both boundary locations and labels. The registration and classification algorithms used by O'Neil-Dunne et al. [12] cannot trivially be extended to produce this information, and we are currently working on a solution.

The added uncertainty information would help with situations such as the Private School candidate, which is a private high school. Due to a combination of poor data quality and gross labeling errors, the school's football stadium is not properly labeled, and the nearby practice field is also missed. The search finds another (incorrect) field, but the resulting attributes make for a poor match. None of the methods can currently handle this case, because none of them account for label uncertainty. We therefore excluded Private School from the list of high schools, as we have no expectation that the search algorithm could systematically find similar examples.

We can also improve the accuracy of the elicited priors by using a more formal elicitation process, such as described by O'Hagan et al. [11]. Separately, while the Latin Hypercube is in theory a good method for selecting informative elicitation points, in practice it tends to create only marginal examples. As a result, the expert gives middling probabilities to all of the elicitation points, and the resulting prior is only mildly informative as evidenced

by the wide uncertainty intervals. A better approach might be to pair the elicitation with an initial search, so that the elicitation points are real candidate matches. This is approximately equivalent to labeling data, but may produce useful models more quickly. In a similar vein, our elicitation efforts revealed that potentially important features, such as proximity to other buildings, were ignored. Elicitation may therefore also need to be paired with query construction, as it is in the distance-based approach.

6 Conclusions

Geospatial semantic graphs offer a new approach to analyzing overhead imagery and other types of data. Our work develops practical uncertainty methods that analysts can use to inform their conclusions. The three methods presented use different information sources, such as elicitation, training data, and distance measures. They also consider different sources of uncertainty, such errors in raw data and initial processing, training examples, and tolerance in the search pattern. Ultimately, all of these need to be accounted for by a single solution while still producing information interpretable by the analyst end-users.

In practice, a hybrid approach may be required. During the initial design of novel queries, labeled examples will be unavailable, and elicitation may be infeasible if the analyst is unsure of how to describe the pattern. The distance-based metrics are most appropriate in this case, as they naturally support an iterative query design process. However, as the search begins to return results that include positive examples, analysts can label data to support the probabilistic models. Ultimately, we expect the probabilistic methods to provide better discrimination, as they implicitly capture from training data aspects of the target pattern that are not captured by the query specification.

Acknowledgments

The authors thank Will Mclendon for his early work on the high school template and for providing the expert elicitation. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at Sandia. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned

subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] E. Bedrick, R. Christensen, and W. Johnson. A new perspective on priors for generalized linear models. *Journal of the American Statistical Association*, 91(436):1450–1460, 1996.

[2] T. Berger-Wolf, J. Berry, S. Bhowmick, E. Casleton, M. Kaiser, V. Leung, D. Nordman, C. Phillips, A. Pinar, D. Robinson, and A. Wilson. Statistically significant relational data mining. Technical Report 2014-1105, Sandia National Laboratories, 2014.

[3] R. C. Brost, W. C. McLendon III, O. Parekh, M. D. Rintoul, D. Strip, and D. M.-k. Woodbridge. A computational framework for ontologically storing and analyzing very large overhead image sets. In *Proceedings of the Third ACM SIGSPATIAL International Workshop on Analytics for Big Geospastial Data*, Dallas, TX, 2014. ACM Press.

[4] R. Carnell. The latin hypercube sampling (lhs) package for R, 2013. URL `http://cran.r-project.org/web/packages/lhs/lhs.pdf`.

[5] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–137, 1997.

[6] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, 1995. Morgan Kaufmann.

[7] S. Kluckner and H. Bischof. Large-scale aerial image interpretation using a redundant semantic classification. In N. Paparoditis, M. Pierrot-Deseilligny, C. Mallet, and O. Tournaire, editors, *IAPRS, Vol. XXXVIII, Part 3B*, pages 66–71, September 2010.

[8] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[9] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Boston, MA, 1997.

[10] H. Nguyen. *Spatial Statistical Data Fusion for Remote Sensing Applications*. PhD thesis, University of California at Los Angeles, Los Angeles, CA, 2009.

[11] A. O'Hagan, C. Buck, A. Daneshkhah, J. Eiser, P. Garthwaite, D. Jenkinson, J. Oakley, and T. Rakow. *Uncertain Judgements: Eliciting Experts' Probabilities*. John Wiley and Sons, Chichester, UK, 2006.

[12] J. P. O'Neil-Dunne, S. W. MacFaden, A. R. Royar, and K. C. Pelletier. An object-based system for lidar data fusion and feature extraction. In *Geocarto International*, pages 1–16. Taylor and Francis Group, 2012.

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan-Kaufmann, San Mateo, CA, 1988.

[14] V. Pronk, S. Gutta, and W. Verhaegh. Incorporating confidence in a naive Bayesian classifier. In L. Ardissono, P. Brna, and A. Mitrovic, editors, *Proceedings of the 10th International Conference on User Modeling, LNAI 3538*, pages 317–326, Edinburgh, Scotland, 2005. Springer-Verlag.

[15] K. M. Simonson, S. M. Drescher Jr., and F. R. Tanner. A statistics-based approach to binary image registration with uncertainty analysis. *IEEE Transations on Pattern Analysis and Machine Intelligence*, 29(1):112–125, 2007.

[16] M. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[17] J.-P. Watson, D. R. Strip, W. C. McLendon III, O. Parekh, C. Diegert, S. Martin, and M. D. Rintoul. Encoding and analyzing aerial imagery using geospatial semantic graphs. Sandia Report SAND2014-1405, Sandia National Laboratories, February 2014.

[18] P. Yue, L. Di, Y. Wei, and W. Han. Intelligent services for discovery of complex geospatial features from remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83:151–164, 2013.