Anomaly Explanation Using Metadata*

Di Qi
Princeton U.
dqi@princeton.edu

Joshua Arfin
North Carolina State U.
josharfin@gmail.com

Mengxue Zhang
Ohio State U.
kikumaru818@gmail.com

Tushar Mathew
AI Software, LLC
tusharmathew@gmail.com

Robert Pless
George Washington U.
robert.pless@gmail.com

Brendan Juba
Washington U. in St. Louis
bjuba@wustl.edu

Abstract

Anomaly detection *is the well-studied task of identifying when data is atypical in some way with respect to its source. In this work, by contrast, we are interested in finding possible descriptions of* what *may be causing anomalies. We propose a new task, attaching semantics drawn from meta-data to a portion of the anomalous examples from some data source. Such a partial description of the anomalous data in terms of the meta-data is useful both because it may help to explain what causes the identified anomalies, and also because it may help to identify the truly unusual examples that defy such simple categorization. This is especially significant when the data set is too large for a human analyst to inspect the anomalies manually. The challenge is that anomalies are, by definition, relatively rare, and so we are seeking to learn a precise characterization of a rare event. We examine algorithms for this task in a webcam domain, generating human-understandable explanations for a pixel-level characterization of anomalies. We find that using a recently proposed algorithm that prioritizes precision over recall, it is possible to attach good descriptions to a moderate fraction of the anomalies in webcam data so long as the data set is fairly large.*

1. Introduction

An anomaly is a pattern or observation that does not conform to expected behavior. The standard anomaly detection task is to identify such anomalies via an algorithm. The anomalies are then subject to further investigation, for example by a human analyst.

But, consider that a peta-scale dataset may have tera-scale anomalies, which are thus beyond the ability of a human analyst to digest. This occurs for example if you set a threshold such as "something that occurs 0.1% of the time is an anomaly." Therefore, the usual approach to anomaly detection is not adequate in that you cannot gather sufficient information about what differentiates these anomalous data points from the rest. We consider the problem of understanding what distinguishes these anomalies. Once anomalies have been detected, we propose a way to find structure within those anomalies to better understand them.

To reiterate, in our scenario, our notion of an "anomaly" is based on a model of expected variations in a data set: we define examples that don't fit this model as anomalies. This may lead to a complex and hard-to-understand anomaly detection procedure. We show how to take such anomaly classifications, and extract a partial, human-understandable characterization of the anomalies.

As an example, suppose you have a collection of traffic camera images, and you want to better understand what comprises atypical traffic conditions. You could obtain a collection of meta-data corresponding to these images. Some attributes might include weather patterns, rush hour periods, and holidays. We propose to consider the task of *identifying what meta-data conditions are associated with anomalous data*. For instance, the co-occurrence of rush hour and snowy weather be correlated in anomalous images, as these conditions could result in traffic delays, low visibility, or accidents. There may be other anomalies that you might be able to identify, but unable to easily explain from a meta-data collection, such as a power outage, or snow covering the camera lens.[1] These, you would perhaps take a separate look at to see if they are relevant. This is the type of application we envision.

In sum, given a list of meta-data attributes about a

*This work was partially performed while D. Qi and J. Arfin were REU students at Washington U. in St. Louis, supported by NSF Award IIS-1560191. M. Zhang, T. Mathew, and R. Pless were also affiliated with Washington U. in St. Louis when part of this work was performed. R. Pless was supported by NSF grant NSF-ISO-1238187. B. Juba was supported by an AFOSR Young Investigator Award.

[1] This last condition can be difficult because it is visually ambiguous, and snowfall is neither necessary nor sufficient for the lens to be covered.

dataset, differences in the meta-data can be used to infer an informative relationship. Our anomaly explanation seeks to attach semantics drawn from the image meta-data to a portion of the anomalous images by returning a list of conditions on meta-data that are associated with anomalies. Not all meta-data attributes necessarily have conditions, as not all are necessarily associated with anomalies.

In this work, we emphasize approximate validity in our descriptions of anomalies. We want the conditions we obtain to reliably indicate that data satisfying those conditions will be anomalous. In this sense, our description of an anomalous event would be informative: when these conditions hold, the image should be an anomaly. Thus, we aim to find a set of anomalies that are easily explainable. These explainable anomalies may the focus of study, or they may be more common anomalies that keep occurring, that a user wants to filter out in order to focus on more interesting cases.

While covering all of the anomalies may also be desirable, this was not the focus of our work. Some anomalies are simply not easily explained, or we may not have meta-data capturing their cause. For instance, the camera could, unbeknownst to us, periodically malfunction. In other cases, it may be that the rule used to make the anomaly classification decision is simply too complex to be captured by a human-understandable rule. In such a case, we inherently must give up on some fraction of the anomaly explanations for the sake of human interpretability. In addition, the unexplainable anomalies could be events that we want to focus on! That our meta-data set is insufficient to explain an anomaly could itself be valuable information.

1.1. Summary of contributions and results

Our main contribution is that we propose a new task, anomaly explanation using meta-data. The key feature of this task is that it is *cross-modal:* we are seeking an explanation of anomalies in one type of data, in this case anomalies in image data, using conditions derived from other types of data. This allows us to connect data with a strong, well-understood semantics such as weather data, the date and time, or image labels, to a pixel-level model of anomalies. We believe such tasks appear well beyond the webcam domain we investigate here, as we discuss in Section 5.

Our second contribution is that we demonstrate that this task can be solved for a standard notion of anomalies in webcam data [16] by using algorithms that prioritize precision over recall, as long as the data set is relatively large, for example, using a training set containing more than 80k images. Given such a large training set, it was able to find simple conditions that explain between 1/10 and 1/6 of the anomalies with precision greater than 75%, and 1/30 of the anomalies with precision greater than 97% (see Section 4 for more detailed results). Using smaller training sets of size closer to 50k, none of the methods we considered, including the baselines, were able to obtain greater than 60% precision, and thus could only address this task weakly at best. We note that since our task is inherently one of learning about events that occur by definition at most 3% of the time, it is not surprising that they should require a large training set (and indeed, also a large test set).

A tertiary contribution is that we conduct the first evaluation of new algorithms, proposed by Juba [11] and Zhang et al. [20], for such high-precision, low-recall tasks on a real-world problem. These algorithms have ironclad but loose worst-case theoretical guarantees, and Zhang et al. presented evidence that their algorithm significantly outperforms the earlier algorithm by Juba using synthetic data. Our results confirm these findings: the algorithm by Zhang et al. performed similarly to the top-performing but harder-to-interpret baseline method (random forest) in the range of interest, and the algorithm by Juba obtained significantly worse results than all of the rest of the methods.

We found that the quality of explanations produced by the meta-data we obtained is fair, but could likely be improved. In particular, many of the image labels we obtained (from Google Vision [2]) were questionable. We investigated omitting these attributes from the data set and found that although a baseline method could still succeed at the task, the human-interpretable methods could not. Thus, the image labels were crucial to the success of the human-interpretable methods we considered. Image labeling was *not* the focus of this work, but our qualitative results suggest that improving the quality of the image labels should substantially improve the quality of our explanations. We leave this to future work.

1.2. Related Work

A number of other works have considered tasks that they refer to as "anomaly explanation," which differ significantly from the task we consider. The main difference is that in our work, the notion of what is an anomaly may be distinct from what is an explanation. Our explanations are given in terms of the meta-data attributes such as time, weather conditions, and so on, whereas the anomaly classification is derived from a model of the distribution of the pixels in an image, which are only indirectly related to the meta-data.

In most other work, by contrast, anomaly explanation involves finding the most influential attributes that caused the anomaly classification rule to classify a given, single observation as an anomaly. For example, Micenková et al. [15] use feature selection methods on a linear classifier to identify the significant features leading to an anomaly classification. Pevny and Kopp [17] simply collect the branches of the trees in a Random Forest [5] that lead to a point being classified as an anomaly to generate a DNF on a per-example basis; Knorr and Ng [12] similarly search directly

to find a minimal set of attributes possessed by an example that cause a point to be classified as an anomaly. Blahut [4] similarly identifies the features of a model obtained via inductive logic programming for "anomalous" points that differ from the models obtained for "ordinary" points. Each of these other techniques would select out "important" pixels to the anomaly classification decision (or worse, a formula indicating that various sets of pixels may be important) which may highlight a region of the image as being significant, but does not solve the task of interpreting what it is about this portion of the image that is anomalous. Furthermore, while explaining the classification of a single point is relevant for some applications, it is a different task from providing a summary of the most common kinds of anomalies produced by the data source, which is the task we address. Since we are most interested in the case where the dataset is very large, it is not clear how to solve our summarization task even given algorithms for the task of explaining why individual examples were classified as anomalies.

Kuo and Davidson [13] similarly solve a variant of the anomaly explanation problem in which, like our problem, a classification of which points are "anomalies" is given as input, and the task is to find a rule that ideally separates the anomalies from the ordinary images. The difference is that for Kuo and Davidson (like Knorr and Ng [12] and the others), this "rule" is actually given by a subspace of the features in which the points labeled as anomalies *are* anomalies in a standard, point-density sense: that is, the points that are labeled as anomalies should have few neighbors in this subspace, and the points that are labeled as ordinary should have many neighbors. So, like in the other variants of the problem (and unlike our problem), ultimately they are identifying a set of features that cause the points to be classified as anomalous in the usual, model-based sense. Also, the density of points with respect to these features must separate *all* of the anomalies from *all* of the ordinary points, whereas we are only seeking to explain a subset of the anomalies.

Babbar [3] casts the problem of anomaly detection and description in terms of Bayesian networks. Here, anomalies are taken to be low probability events based on the joint probability distribution, after which they are scored to determine if they are "genuine" or "trivial" anomalies based on a user defined cutoff. As with most other methods discussed earlier, this method also aims at picking out the attributes which contribute to the standard anomaly classification; again, this is unlike the method we are proposing. We note that one could set up a larger Bayes net that encompassed the meta-data attributes as well, and select out sets of attributes that have near-zero probability conditioned on a non-anomalous classification. This would address our task, and the key question with such an approach is how well it can be made to scale. But again, we stress that this was not the approach considered by Babbar.

Lastly, it should be noted that in our experiments, although we will use Random Forest as a baseline, we will *not* use Pevny and Kopp's actual Random Forest-based method for the individual examples. As mentioned earlier, the objectives in their work are slightly different as they are looking at explaining a specific point as an anomaly, whereas we want to characterize a large class of anomalies.

2. Dataset

The data combines webcam image data with meta-data collected from a variety of sources. These meta-data concern the semantic contents of the image (i.e., image labels obtained via object recognition) and local conditions when the image was collected.

We use webcam data from four locations in the AMOS database of webcams [9], which takes a photo from each webcam approximately every thirty minutes. We selected these locations based on three criteria: first, they are very stable; second, they have a relatively large number of images for the AMOS collection; and third, they are located in the USA, and so data on the weather and holidays at these locations was readily available from common sources. For each camera, there were between 73847 and 131873 images, and the meta-data ranged from 195 to 335 dimensions. The locations are a pond (camera #269), Lake Mono (#4312), the Moody Gardens theme park (#623), and a Toledo highway (#21656). The actual longitude and latitude of the cameras have been estimated using prior work by Jacobs et al. [10].

To gather meta-data, we use the Google Vision API [2], the SunCalc API [1], the Python holidays library [18], and data from The Weather Channel [19] to generate a binary and nonbinary meta-data collection. The binary version is intended to provide a "summary" of nonbinary variables for those methods that require all variables to be binary. The scripts we used and the meta-data apart from the Weather Channel data is available at bitbucket.org/pastateam/pasta. (The Weather Channel data is available for purchase from the company.)

We use the label detection feature in the Google Vision API to obtain a list of objects found in images. Each label was a variable. If an image did not receive that label, then the variable takes the value 0. If it received the label, the variable takes the value 1 in the binary dataset, and the value of the score in the nonbinary dataset. The score is a value between 0 and 1 that represents the confidence that the label is relevant to the image. Google Vision failed to run on a small number of the images. For these, we mark all Google Vision variables as 0 and for all images, we append an additional variable to signify whether or not Google Vision successfully ran.

We use the SunCalc API to create a list of binary vari-

ables for each image based on whether or not it was taken within a certain sunlight phase, for example, during sunrise. We use data from The Weather Channel to generate binary and nonbinary variables based on weather phenomena, including precipitation and cloud cover. Lastly, we include binary labels for weekends and holidays.

Figure 1 shows a sample of the meta-data labels generated for an image from camera #269. Note that we also included the absence of a particular label that was found in other images as a feature, in this case the absence of the labels "weekend", "freezing", and "coast." A complete list of the meta-data attributes we used for each camera is included in the bitbucket repository.



dry, sea, humid, not weekend, landscape, cloud, not freezing, plain, mountain, lake, cloud, not coast, reservoir, sky, ...

Figure 1: A subset of meta-data labels for an image from camera #4312

3. Procedure

We considered anomaly classifications produced by principal component analysis (PCA). Following prior work [16], we defined anomaly scores by the reconstruction error using three principal components, and defined the top 3% scoring images as anomalies. We confirmed that there was no significant improvement to using five, ten, or twenty components, nor was there any significant change to our results when defining 1%, 2%, 4%, or 5% of images as anomalies. For time intervals where the camera changed resolutions or moved, we obtained independent PCA decompositions. These changes would cause predictable reconstruction errors that would have detracted from the goal of finding a diverse set of meta-data explanations.

We evaluated methods of producing rules that select a subset of the images from the camera based solely on the meta-data. We can think of this selection as a classification of images as anomalous or not, that is allowed to make false-negative predictions but (ideally) not false-positive predictions. Thus, the key metrics we considered were the *precision* and *coverage* of the explanations generated, where the coverage simply refers to the proportion of images selected. If the precision is high, the coverage is essentially an unnormalized version of the recall. (That is, we can essentially obtain the recall by dividing by 3%, which by definition is the proportion of the data set that is classified as an anomaly.) Thus, 100% precision and 3% coverage is ideal.

We evaluated four algorithms for this task: the Patient Rule Induction Method (PRIM) [8, 7, 6], random forests [5, 14], and two new algorithms from the artificial intelligence community that prioritize precision over recall, and thus were good candidates for this task. We used standard implementations of PRIM and random forest, and the implementations of the new methods are available at bitbucket.org/pastateam/pasta and github.com/kikumaru818/RedblueSetCover.

The first of the new methods, Tolerant Elimination, was introduced by Juba [11]. The algorithm seeks to find a k-DNF explanation for some fixed k, i.e., an OR of ANDs of k "literals"—our Boolean attributes or their negations. It forms a working hypothesis that initially includes all possible terms of size k. It then tries to narrow this k-DNF down to the best definition by iteratively eliminating terms that have more false-positives than a bound calculated based on a user defined tolerance parameter and the predicted probability of hitting an outlier. The algorithm iteratively reduces the target coverage bound until it either finds a formula that approximately achieves the target coverage, or determines that the bound is too small for statistical validity. These tight bounds restrict the algorithm, leading it to report a very high error rate in comparison to the other algorithms used for evaluation on most datasets because it was not able to eliminate enough terms.

Low-degree Greedy Cover is the second new algorithm, due to Zhang et al. [20], for the same task as tolerant elimination. This algorithm finds a k-DNF with high precision by iteratively ignoring terms that exceed a given false-positive threshold, ignoring points that are false-positives for too many terms (using a corresponding threshold), and using a greedy selection of terms to classify a specified fraction of the points as positive. By ignoring the terms and points that may "share too much," the errors due to each term can be treated roughly as fixed costs: by carefully choosing these thresholds, they find that the new method has quadratically smaller error than Tolerant Elimination in the worst case. Moreover, they found that this new algorithm substantially outperformed Tolerant Elimination on a synthetic benchmark. To our knowledge, neither algorithm had been used for any real-world problem until the present work.

In both of these methods, the size of the terms, k, is a parameter that controls a trade-off between the expressive power of the explanations we can generate, and the running time and amount of data needed. Juba and Zhang et al. provide theoretical guarantees for their effectiveness at the statistical task, with running time and data requirements both growing exponentially with k. Due to the running time in practice, it was difficult to scale these methods beyond $k = 2$ while using the full set of attributes, and so our experiments only consider $k = 1$ and $k = 2$.

PRIM [8, 7] is another well-known method for such statistical tasks that produces essentially a DNF as output (and is thus interpretable), although it does not have the same guarantees. Specifically, it finds a region of the input associated with high-valued values of a dependent output variable. We ran PRIM once with the real-valued meta-data and PCA reconstruction errors and another time with real-valued meta-data and binary representations of anomalies. PRIM has a peeling step to remove non-anomalous data and an optional pasting step to correct for over-peeling. Because pasting failed to complete on a cluster with a 2.3 GHz processor and 192GB RAM after 14 days, we omitted this step.

We also felt it was useful to compare these methods to some established state-of-the-art baseline method for such high-precision classification without regard for interpretability. We chose Random Forests [5] since this is a well-known baseline that has seen use as an input for other kinds of anomaly explanation, and thus we had reason to believe it would be effective. Specifically, Pevny and Kopp [17] use Random Forests to produce per-example explanations of anomalies, in contrast to our explanations that describe conditions that capture the variety of anomalies observed on a given camera. We simply used the standard Random Forest classifier as a baseline, which we anticipate to be more accurate but surely less interpretable than the actual method used by Pevny and Kopp. For this method we used binary representation of both meta-data and anomalies.

4. Results

We evaluated the four methods described in the previous section on our four data sets using a three-fold cross-validation. Thus, for each fold we used 2/3 of the data set, including both anomalous and non-anomalous data, for training; used the remaining 1/3 as a test set; and averaged the three results.

4.1. Semantic Explanations

We draw some examples of explanations from the pond location (camera #269). Two non-anomalous images from this camera are shown in Figure 2.

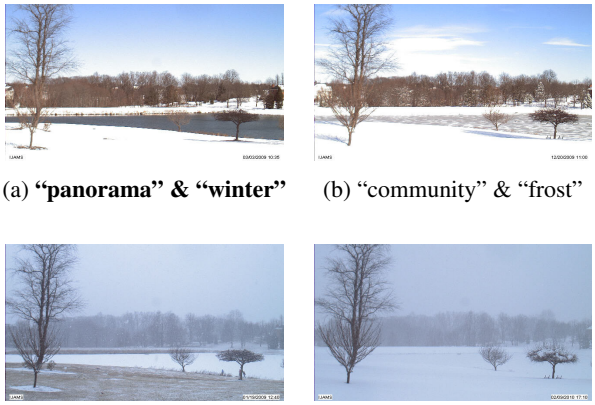
We first examine the terms generated as explanations of anomalies by the algorithm of Zhang et al. [20] run using 2-DNF with 0.3% coverage; it obtained a precision error of



(a) 4.37th percentile (b) 92.91st percentile

Figure 2: Non-anomalous images for camera #269 with percentile score of its PCA reconstruction error. The images with anomaly scores greater than 97 percentile are classified as "anomalies."

1.5% (i.e., out of the points satisfying the condition returned by the algorithm, only 1.5% were *not* anomalies). Since the data set only contains 3% anomalies, this condition captures essentially 1/10 of the anomalies. The four terms describing the condition, together with examples of images satisfying those individual terms, appear in Figure 3.



(a) **"panorama" & "winter"** (b) "community" & "frost"

(c) "rain and snow" & "holiday" (d) "branch" & "dry"

Figure 3: Terms covering 0.3% of images (1/10 of the anomalies) with 1.5% test precision error for camera #269, illustrated by example images. The term in bold, 3a, was contained in all of the three cross-validation runs.

We can see in Figure 3 that all three cross-validation runs contained the term "panorama" and "winter," depicted in Figure 3a. Indeed, this single term achieved 0.2% coverage (1/15 of the anomalies) and test precision error was approximately 0%. Four more images satisfying this term can be seen below, in Figure 4. We believe that this is a reasonable description of these anomalous, snow-covered scenes.

Indeed, three out of four of the terms in Figure 3 reasonably describe the anomalous image, but one, "branch" and "dry" seen in 3d, is not so reasonable. (Further examples of images labeled by the other two terms are available at bitbucket.org/pastateam/pasta.) All 62 images that had these



Figure 4: A subset of the 235 images from camera #269 that contain the terms "panorama" and "winter," a 2-DNF explanation generated by the algorithm of Zhang et al. [20].

labels were marked as anomalies, so from a purely quantitative standpoint, this tag is quite good. Four of these images can be seen below, in Figure 5. A further 16 of these images are available at bitbucket.org/pastateam/pasta. These images seem to be distinguished by a flat white landscape that gives no sense of depth, perhaps leading the tree to be mislabeled as a "branch."



Figure 5: A subset of the 62 images from camera #269 that contain the terms "branch" and "dry," a 2-DNF explanation generated by the algorithm of Zhang et al. [20].

The 36 terms generated by the algorithm of Zhang et al. [20] for 0.5% coverage contain the terms that appeared in 0.3% coverage, in Figure 3, together with the terms displayed in the Supplementary Material at bitbucket.org/pastateam/pasta (Figures 1 and 2). In this scenario, most of the scenes detected as anomalous were winter scenes or image encoding errors. The labels largely reflect this: many of the terms selected, like those illustrated in Figure 3, include terms referring to snow, ice, or winter. Similarly, many of those capturing an image encoding error do give a rough characterization of the error: for example, the labels "purple" and "yellow" capture, respectively, images that have a purple and yellow tint due to encoding errors; meanwhile, labels like "texture" or "macro photography" capture images that are heavily blurred. But, as previously mentioned, the quality of explanations was negatively affected by image label quality. For example, for this camera, we see labels such as "farmhouse" and "ice boat" that do not seem to refer to anything that appears in the image.

We also examine the terms obtained for camera #623, the Moody Gardens theme park. A couple of ordinary images for this camera are shown in Figure 6.



(a) 91.42nd percentile (b) 21.26th percentile

Figure 6: Non-anomalous images for camera #623 with percentile score of its PCA reconstruction error.

The 2-DNF explaining 0.1% of the images (1/30 of the anomalies) for camera #623 is shown in Figure 7. Although we were able to obtain a similarly high rate of precision for covering 0.1% of the images (2.2% test precision error), as a consequence of the quality of the image labels, the anomalies are harder to interpret for this camera. For example, the image in Figure 7c is actually an anomaly because the lamps in the parking lot are not lit, but the image has been labeled by the unusual term "gadget" that has no clear relationship to the image's contents. (We note that this term did not appear in some of the cross-validation runs.) Further examples of images labeled by all four of these terms are included in the Supplementary Material at bitbucket.org/pastateam/pasta. The formulas covering 0.3%–0.5% of the images contained 84 and 98 terms, respectively, and are not included here.

For the remaining two cameras, as we will discuss in more detail in the next section, no method (including the baselines) could obtain a satisfactorily high-precision characterization of the anomalies given the data available. We believe that this is due to the training set being too small. These cameras also suffered from poor image label quality; for example, in the Toledo highway location (camera #21656), a truck was mistakenly identified as a "jet aircraft." Thus, we found that the overall quality of the semantic explanations was poor for these cameras. We have included some example images for these cameras at bitbucket.org/pastateam/pasta, but we do not further discuss the contents of the rules found to describe the anomalous images for these cameras.



(a) **"text" & "image"**



(b) **"bird's eye view" & "peninsula"**



(c) "gadget" & "weekend"



(d) **"horizon" & "bird's eye view"**

Figure 7: Terms covering 0.1% of images (1/30 of the anomalies) with 2.2% test precision error for camera #623, illustrated by example images. The terms contained in all of the three cross-validation runs are in bold.

4.2. Quantitative Comparison of Methods

The performance of the methods on the four data sets are shown in Figure 8. Actually, the performance of Juba's [11] Tolerant Elimination algorithm is not included in these graphs. It had an error rate ranging from 92% to 98% for all cameras and anomaly percentages, which was significantly worse than what could be achieved using the other three methods.

We stress that by definition, the data only contains 3% anomalies. Thus, any classifier that covers, say, 6% of the data necessarily has a precision error of greater than 50%.

Camera	Precision error with 1% coverage	Precision error with 3% coverage
#623	0.3930	0.7713
#269	0.1366	0.4082
#4312	0.7603	0.8521
#21656	0.8579	0.9110

Table 1: Precision error rates obtained by Low-degree Greedy Cover with 3% of elements defined as anomalies

For this task it appears that having a larger data set is extremely important for generating a set of explanations with low precision error rates. For instance, cameras #623 and #269 have 131873 and 123886 images respectively (Figure 8a,b), whereas cameras #4312 and #21656 have 75453 and 73847 images respectively (Figure 8c,d), and the latter

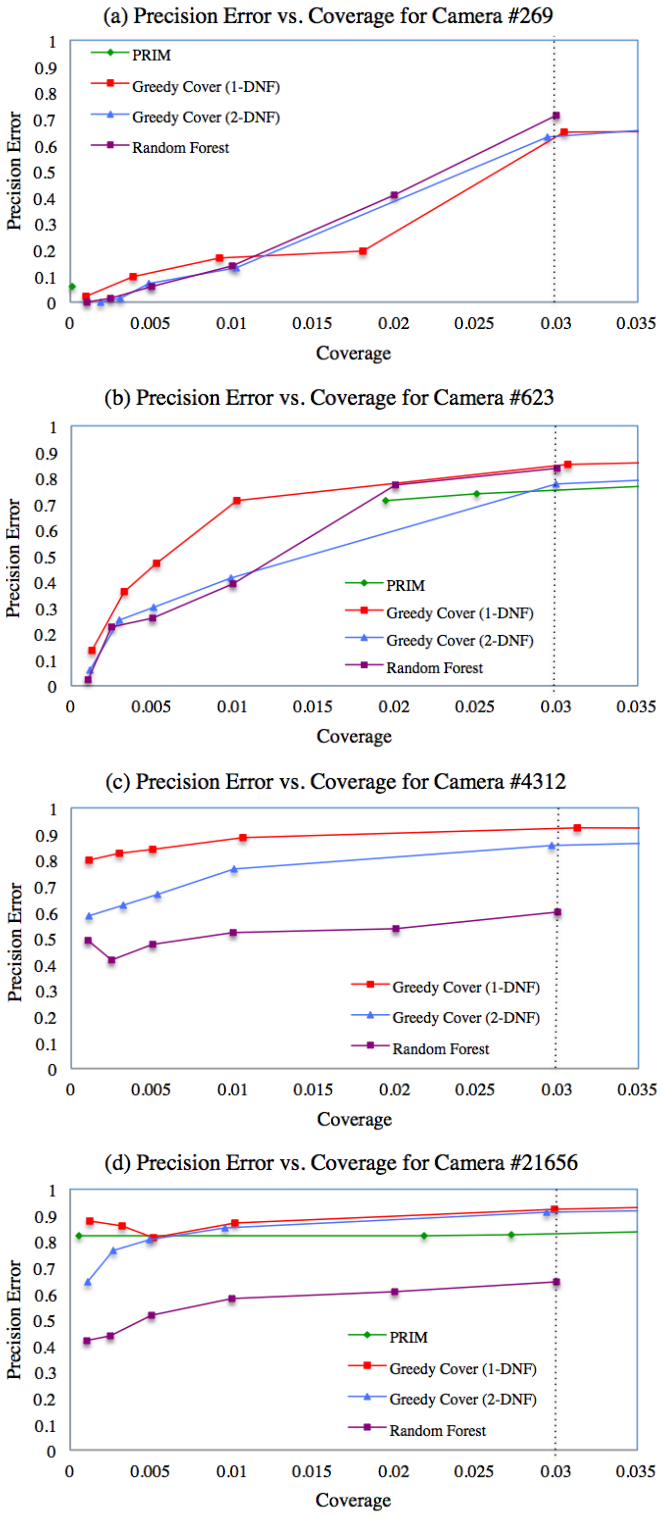


Figure 8: Precision error vs. coverage plots. (a) and (b) are larger datasets with >80k training examples; (c) and (d) are smaller datasets with ≤50k training examples. Observe that no method obtains precision substantially greater than 60% for the smaller datasets, while for the larger datasets we can obtain greater than 75% precision as long as we only fit a fraction of the anomalies.

have higher error rates as shown in Table 1 for the Low-degree Greedy Cover algorithm [20] and in the precision error vs. coverage graphs. Moreover, for smaller coverage rates, the plots show that the algorithms are overfitting, as evidenced by the increase in precision error as the target coverage decreases. Note that in principle, a smaller coverage should lead to a lower precision error, so the increase in error rates cannot be inherent to the task. That such error might occur due to overfitting is unsurprising: for cameras #4312 and #21656, 1/30 of the anomalies correspond, respectively, to 51 and 50 images in the training set. We certainly do not expect reliable statistical estimation from such small numbers of positive examples, and on the contrary we expect overfitting should occur. In any case we observe that for the larger data sets, it was possible to obtain sufficiently high precision to obtain high confidence that the rules we found really do indicate that an image will be an anomaly, as long as we aim to explain a moderately small fraction of the possible anomalies.

Although Figure 8 demonstrates that Low-degree Greedy Cover is quantitatively competitive with random forest on cameras #269 and #623 where sufficient data was available, we were occasionally disappointed by the quality of the explanations using attributes produced by Google Vision when these contained spurious tags. We thus also evaluated the performance of the methods on the four data sets omitting the labels obtained from Google Vision, seen in Figure 9. We were interested in whether or not explanations relying on the better grounded data about the weather, time of day, and so on could provide higher-quality explanations. The performance of our baseline method, Random Forest, demonstrates that there is indeed enough information in the meta-data to perform the task reasonably well. But, unfortunately, the human-understandable methods did not succeed at extracting the relevant information from these attributes. (Plots for cameras #4312 and #21656, again reflecting the lack of training data can be found at bitbucket.org/pastateam/pasta.) It is an interesting open question whether or not some other human-interpretable method can succeed in this setting. We leave this to future work.

5. Extensions and Further Applications

We believe that this task has applications well beyond the webcam domain we investigated here. For example, there are well-known models of typical data (and hence, also anomalies) in time-series data such as stock market indices. In this context, supposing we are given a source of meta-data such as the day's news articles or messages on a social network from the previous hour, we could attempt to produce a summary of what events may have corresponded to anomalously large (or small) stock movements in terms of keywords or topics appearing in these sources. Note that the stock price changes themselves have a rather limited semantics, and thus the meta-data is essential for such a task. In the future, we hope to investigate such further applications.

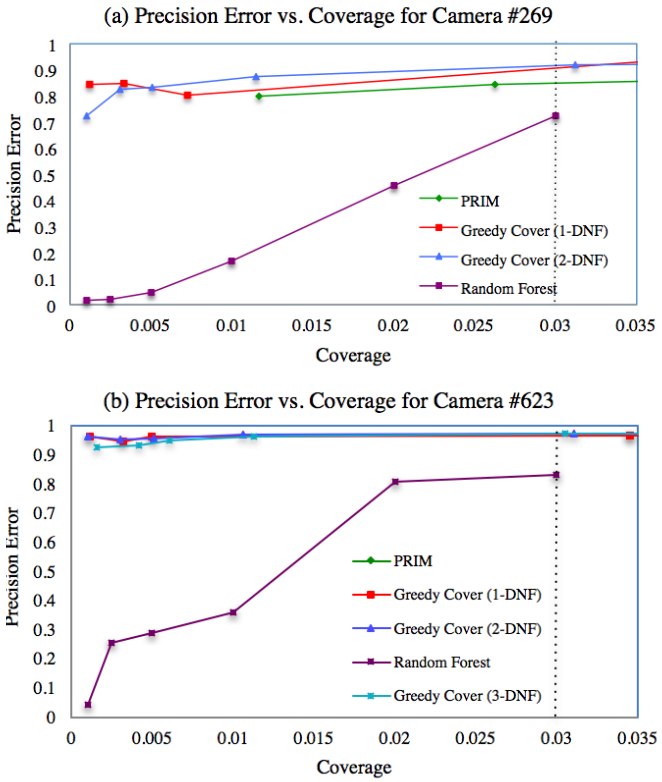


Figure 9: Precision error vs. coverage plots for data sets omitting Google Vision data for cameras #269 and #623. The Random Forest baseline can achieve low precision error but none of the other, interpretable methods do.

Acknowledgements

We thank our reviewers for their constructive comments and suggestions.

References

[1] V. Agafonkin. Suncalc api. https://github.com/mourner/suncalc, 2015.

[2] Alphabet. Google vision api. https://cloud.google.com/vision/, 2016.

[3] S. Babbar. Detecting and describing non-trivial outliers using bayesian networks. In *Proc. Cognitive Computing and Information Processing (CCIP)*, pages 211–222, 2015.

[4] V. Bahut. Outlier detection and explanation. Bachelor's thesis, Faculty of Informatics, Masaryk University, 2015.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] J.-E. Dazard, M. Choe, M. LeBlanc, and J. Rao. R package primsrc: Bump hunting by patient rule induction method for

survival, regression and classification. In *JSM Proceedings. Section for Statistical Programmers and Analysts*. American Statistical Association-IMS, Seattle, WA, USA, 2015.

[7] J.-E. Dazard and J. Rao. Local sparse bump hunting. *J. Comp Graph. Statistics*, 19(4):900–929, 2010.

[8] J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

[9] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007.

[10] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–6, 2007.

[11] B. Juba. Learning abductive reasoning using random examples. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 999–1007, 2016.

[12] E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers. In *Proc. Very Large Data Bases Conference (VLDB)*, pages 211–222, 1999.

[13] C.-T. Kuo and I. Davidson. A framework for outlier description using constraint programming. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1237–1243, 2016.

[14] A. Liaw and M. Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[15] B. Micenková, X.-H. Dang, I. Assent, and R. T. Ng. Explaning outliers by subspace separability. In *Proc. International Conference on Data Mining*, pages 518–527, 2013.

[16] J. O'Sullivan, A. Stylianou, and R. Pless. Democratizing the visualization of 500 million webcam images. In *Applied Imagery Pattern Recognition Workshop (AIPR)*, 2014.

[17] T. Pevny and M. Kopp. Explaining anomalies with sapling random forests. *Information Technologies - Applications and Theory Workshops, Posters, and Tutorials*, 2014.

[18] ryanss. Holidays library. https://pypi.python.org/pypi/holidays.

[19] The Weather Channel. https://weather.com.

[20] M. Zhang, T. Mathew, and B. Juba. An improved algorithm for learning to perform exception-tolerant abduction. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 1257–1265, 2017.