A Framework for Analyst Focus from Computed Significance

D. B. Skillicorn

July 14, 2010

Abstract

Attention is the critical resource for intelligence analysts, so tools that provide focus are useful. One way to determine focus is by computing significance. In the context of a known model, new data can be placed on a spectrum defined by: normal, anomalous, interesting, novel, or random; and significance is greatest towards the middle of this spectrum. However, significance also depends on the mental state of the analyst (and the organization). A framework for understanding significance is defined, and its impact on the knowledge discovery process is explored.

1 Motivation

In adversarial knowledge discovery settings, such as counterterrorism, counterintelligence, law enforcement, fraud detection, financial tracking, and so on, analyst/investigator attention is the critical resource. These settings are examples of what Treverton [17, 18] calls *mysteries*, issues where framing the right questions is important and it is the quality of the analysis, rather than the amount of data, that is critical (in contrast to *puzzles* where questions can always be answered given enough data).

Tools that can direct focus to the most significant aspects of data, and the models built from data, improve effectiveness, and reduce timelines. In high profile terrorism cases, for example the attempt by Umar Farouk Abdulmutallab to blow up a plane at the end of 2009, it often turns out that the data necessary to detect and prevent the attack had been collected, but insufficient analysis had taken place to enable detection beforehand. In other words, the problem is not so much "connecting the dots" as finding those dots that are important enough that their connections need consideration in a pool of dots that may be nine orders of magnitude larger. The biggest win, therefore, in tool design and implementation is improvement in the collaboration between analyst attention and tool-directed focus. This will obviously be easiest when tools exploit human strengths, for example, using the high bandwidth of the human visual system by generating effective visualizations, or acknowledging the small size of human working memory by keeping focus small.

In adversarial settings, analysts and tools need to be aware of the implicit 'arms race' between adversaries' attempts to conceal themselves and manipulate the analysis process, and the quality,

toughness, and timeliness of models. This implies that models will be updated regularly, and that update is a far more integral part of the process than typical business modelling. In particular, significance is a moving target, even independent of changes in the data or the analyst's mindset or knowledge.

A problem is that significance is not a property only of the data or models in themselves. It does not make sense to talk about a significant record, or a significant decision boundary, or a significant cluster without some context and usually some history. Significance is a relation between data or model on the one hand, and analyst state and history on the other. Indeed, what may be significant to one analyst may not be to another, depending on how much they already know and understand. Therefore, tools that use significance as a way to generate focus can only do well when they include some input that reflects the analyst/organization worldview.

There is, however, a danger of overshooting when analyst state is included – analysts can misinterpret new signals as something they already understand, when this is not the case. Another frequent refrain in terrorist attack post mortems is that there has been a "failure of imagination", essentially a claim that analyst mindset ruled out some possibilities. This can work in two ways: data or models are discounted as too unlikely, or discounted as something already understood; but, in both cases, they are discounted. It is important that tools using significance should be resistive to under- and over-reliance on analyst state; they should include it, but cautiously.

The contribution of this paper is to define a framework for significance, define how significance depends on the relationship between new data and existing models, and to explore how knowledge-discovery techniques can integrate the idea of significance and analyst context. As usual, some techniques are significance-ready, requiring only that some of their existing functionality be better used; others require substantial enhancement.

2 The Structure of Significance

Models that aim to find the "interesting" or "anomalous" or "novel" data or parts of a model often conceive of the problem, perhaps implicitly, as fundamentally two-class – objects are either normal or anomalous, known or new. I suggest that there is more structure to the problem, and that acknowledging and incorporating this structure provides a better way to design systems to recognize significance.

Every model is imperfect because it was built from less data than could have been collected and used, and because the model-building algorithm was restricted in form and so in power. The arrival of new data may reveal either or both of these limitations of the current model.

Figure 1 shows a very simple model in two dimensions, with records shown as points; implicitly, distance represents dissimilarity. The obvious structure of the model is three clusters, indicated by labelling the points with different symbols. Now consider new records whose attributes place them in the positions indicated by the lower-case letters. Each record is new and therefore potentially represents significant new knowledge to which an analyst might need to pay attention. However, these points are qualitatively different from each other, and it is this qualitative difference, I contend,

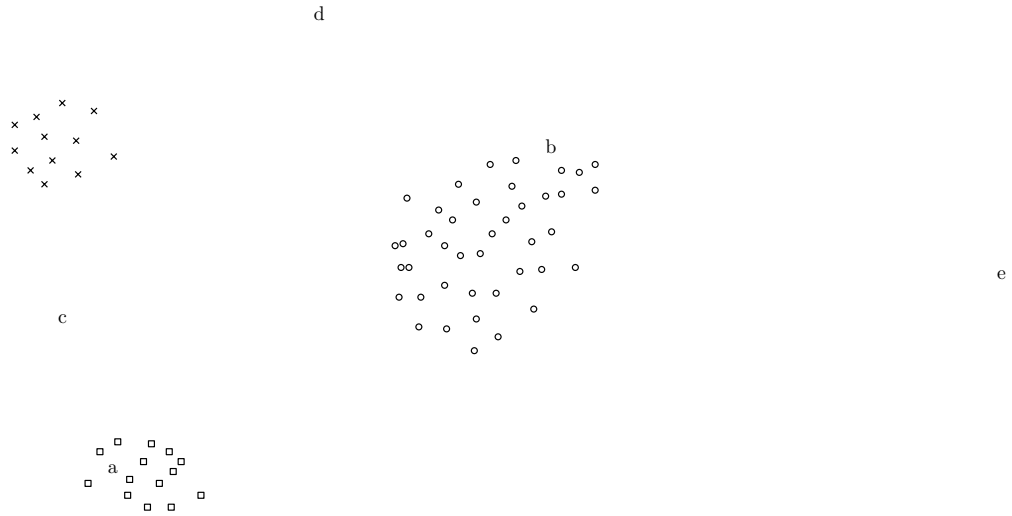

Figure 1: Different possibilities for significant data in a simple dataset. Point a is ordinary; point b is anomalous; point c is interesting; point d is novel; and point e is random.

that is the structure of significance. Their meanings are as follows:

- Point a is fully explained by the existing model – although it is new, it is not significant.
- Point b is *anomalous*; it lies just outside one of the existing clusters but is so close to it that it exerts no pressure for a new understanding. Its presence is explainable by the finiteness of the particular sample that was used to build the model (the clusters). Each cluster might be considered to be wrapped by a boundary that falls just outside the points within it, and an anomalous point lies within this boundary.
- Point c is *interesting* – its presence suggests a weakness in the current model, perhaps the presence of a yet-unrecognized fourth cluster, or the fact that the square cluster and the cross cluster are actually the same. The entire set of clusters might be considered to be wrapped by a boundary just outside the set of cluster boundaries. An interesting point lies within this boundary but not close to any existing cluster.
- Point d is *novel* – its presence does not indicate a weakness with the model as such because it is well outside the range of data from which the model was built. Its presence does, however, require it to be acknowledged and treated using a different process (which could be as simple as ignoring it, although this might be dangerous). A novel point lies outside the boundary wrapping the clusters.
- Point e is noise – its presence brings no new information because it is so far outside of the situation being modelled. Another, much more amorphous, boundary defines the difference between novel and noisy, perhaps constructed by considering multiples of the standard deviation of the entire dataset.

Of course, these categories and their boundaries are not hard and fast, and new points might be hard to unambiguously place in one or another category. Nevertheless, it seems helpful to consider them as qualitatively different when deciding what each *means* and what should be done as a result of the arrival of a new record of each kind.

These categories form a natural spectrum: *normal* – *anomalous* – *interesting* – *novel* – *random*, but what makes dealing with this technically difficult is that significance peaks in the middle of the spectrum. New data that is normal or random is not very significant, new data that is anomalous or novel is moderately significant, but data that is interesting is most significant of all. As the figure shows, it is relatively straightforward to detect data that is normal or random, but this becomes more difficult, the closer the data is towards the center of the spectrum. Not every technique for finding and displaying significance is congruent with this structure, but the more they are, the more effectively they work. There are connections here to, for example, intrusion detection, but in that setting all data that is not normal is lumped together, and the issue of inadequacy of the model is not usually considered.

3 The Role of Context

Most knowledge discovery operates either on record-based data, or graph-structured data. Record-based data consists of a set of n records, each with m attribute values and so has a natural geometric representation in m-dimensional space, where each attribute is associated with a dimension, and each record with a point. Graph-based data consists of n records, each regarded as a graph node, and (weighted) edges connecting some pairs of nodes. The advantage of graph representations is that they integrate the data globally (the structure depends on every edge) – this makes them resistant to adversarial manipulation, because many of their properties are emergent, but they are practically more difficult to work with.

Record-based data can be, and often is, converted to graph-structured data by constructing a local similarity between every pair of records. Local similarity may be derived from correlation between records, or Euclidean distance in the geometry. Local similarity is often thresholded, set to zero for records that are insufficiently correlated, too far apart, or not among each other's k nearest neighbors.

The significance of a record or a part of a model is usually not just a function of the model, but also of the context in which that model is being considered. In other words, significance is a stateful property and depends on what is already known and understood, and the (mental) weights assigned to each part of the model. This background state includes not just the mental state of each analyst but also aspects of the state of the entire organization.

There are connections here to work on e-learning, where individual learners interact with online material to 'teach themselves'. Building such systems requires not only presenting material, but creating different, and consistent, paths through the material to suit different learning styles (for example, [19]). It also requires building a model of each learner to predict which path, and which next step, should be presented. Unfortunately, this work is not directly applicable to intelligence analysis because it relies on knowledge of the total content to be learned, often captured by an

ontology, and on large-scale commonalities across learners. In other words, e-learning focuses on learning concepts that are understood by those building the system, and the learners are all intended to reach the same state; whereas in intelligence and law enforcement, the outcome is not yet known by anyone, and different analysts may reach different conclusions. E-learning is about solving a puzzle; most intelligence and law-enforcement analysis is about better framing a mystery.

There are three ways of addressing analyst context in significance computation:

1. Ignore context. Such systems implicitly answer the question "show me what is significant" and all users get the same answer.
2. A classical-physics view, in which the model underneath is not affected by the context, but the rendering, in its most general sense, is. Significance computation acts as a filter on the unchanging underlying model. Such systems implicitly answer the question "given what I know, show me what is significant (to me)".
3. A quantum-physics view, in which the model is reconstructed to reflect the analyst context. Such systems implicitly answer the question "using what I know to recalibrate the model, show me what is significant to me".

These methods are listed in increasing order of effectiveness, but also in increasing order of difficulty.

Of course, another way in which context plays a role in model building is by the inclusion of *domain knowledge*. Such knowledge is helpful to constrain the kinds of models that are built, eliminate models or parts of models that are not helpful, focus attention on models or parts of models that are likely to be particular revealing, and guide the collection and use of high-value attributes. However, this aspect of context is usually taken into account at an earlier stage and by different people than analysts.

3.1 Non-Contextual Modelling

Many existing knowledge-discovery techniques provided some kind of significance indication, often as a side-effect. For example, some predictors (support vector machines, ensembles, random forests, some neural networks) provide not just a class label prediction but also some indication of how robust that prediction is. Predictions close to decision boundaries are *interesting*. On the other hand, predictors are poor at detecting *novel* records – because these are mostly far from decision boundaries they tend to be classified with high confidence, despite being unlike any of the data on which the predictor was trained.

Clustering algorithms also sometimes provide extra information. Distance-based clustering can identify records that are far from all clusters (*novel*) and equidistant from all/some clusters (*interesting*). Density-based clustering can identify records that are not close to any cluster, but has a harder time distinguishing *novel* from *interesting*.

Predictors and clustering algorithms can easily be improved by routinely adding an ancillary 1-class clustering algorithm whose role is to examine new records and decide whether or not they

are *novel*. This could be done with differing levels of sophistication, ranging from wrapping the 'known' data region in a convex hull, to using a 1-class SVM [16]. This would prevent predictors silently making predictions for records unlike any they were trained on; and makes it easier to identify *interesting* records in clusterings [2] – they are far from existing clusters but not *novel*.

Determining significance without any context means that the process must be completely inductive. A useful property of significance is that, roughly speaking, frequent records cannot be significant for two reasons: first, adversaries tend to be rare, so records of their actions are not usually common; and, second, common records tend to be well accounted for by basic models. This underlies some of the mechanisms outlined above.

However, the uncommon records or parts of the model that represent them are not necessarily the most significant, and this is the technical challenge. Uncommon records are a mixture of *interesting*, *novel*, and *random* records, and perhaps records that reflect individual eccentricity. Separating these categories is not always straightforward.

A major issue is that, using only induction, the relative significance of different data depends heavily on how the values of the data are normalized. In general, attribute values are expressed in units that are not naturally comparable across attributes; and where the relationship between magnitude and importance is not linear for each individual attribute. Without care, differences in significance can be artifacts of normalization choices – but there are seldom enough principled reasons for definitively choosing one normalization over the others.

A number of knowledge-discovery tools address significance explicitly. The first, and most obvious, is Pagerank [4, 5], the algorithm used by Google to rank web pages. This is an example of a general approach based on ranking relative to the first eigenvector of a matrix.

Given a graph with positive weights on the edges, the first eigenvector of the adjacency matrix points from the origin towards nodes that are well-connected by high-weight edges. Projecting all nodes onto this vector generates a ranking where high significance is identified with projection far from the origin.

If significance actually is a single-factor property, then this works well. However, if it is not, it is difficult to extend to idea to include more factors. The first eigenvector passes from the origin through a hypercube in the positive hyperquadrant towards the centroid of the data. The second eigenvector is constrained to be orthogonal to the first, but this is not necessarily the direction of maximal remaining variation in the data.

The problem can be avoided, if necessary, by normalization. For record-based data, normalization that shifts the set of points so that the centroid is at the origin removes the misleading, and often useless, initial vector [15]; for graph-based data, replacing the adjacency matrix by one of the graph Laplacians does the same thing [20]. After these normalizations, an eigendecomposition or singular value decomposition can be truncated after some chosen k dimensions, projecting the data into a k-dimensional space where its structure is more easily seen. If a specific significance ranking is needed, points can be projected onto the vector passing through the point $(\sigma_1, \sigma_2, \ldots, \sigma_k)$. Extremal points in this ranking may be *random*, *novel*, or *interesting* depending on what other processing was done beforehand. However, the distributions of points along this line may itself

inductively provide information on which category to assign each point to. For example, a small but modestly outlying cluster of points in the ranking is probably *interesting* rather than *novel*, because correlated unusual activity is a strong signal of adversaries.

For graph-based data, the analysis possibilities are richer. Suppose, for simplicity, that the graph has n nodes and is connected. The graph Laplacian is obtained from a weighted adjacency matrix by computing the row sums, replacing off-diagonal non-zero entries by the negation of their values divided by the row sum, and replacing the diagonal entries by 1's [1]. After an SVD of this matrix, the main structure is revealed in columns $n-1, n-2, \ldots$ of the U matrix – for example, plotting the points using these two columns provides a good drawing of the graph and can be used to partition it into subclusters. In particular, sorting the nodes by the magnitudes of the entries in column $n-1$ provides a global importance ranking roughly equivalent to that of Pagerank.

However, the SVD does not know that the initial matrix describes a graph, so information about variation is also captured in columns $1, 2, \ldots$. Ranking nodes by the magnitudes of the entries in column 1 provides a ranking by how unusual the local neighborhood of each node is, in particular how well it is connected to its neighbors.

Thus the columns at one end of the distribution provide information about the 'big' structure of the graph, and so which nodes are significant in the context of the large-scale structure of the graph (its clusters). The columns at the other end provide information about the local structure of the graph, and so which nodes are significant in the context of small-scale structure.

But wait, there's more. The columns in the 'middle' of the U matrix (where 'middle' depends on the precise structure of the graph, but can be found by looking at the absolute value of the sums of columns) reveal small, unusual regions of the graph [14]. Most of the values of these columns are zero; the non-zero entries provide information about the strong nodal domains which are made up of nodes from which the view of the rest of the graph is unusual [6].

If the graph represents relationship among people, one set of columns provides a view that focuses on power and who is leading the group; another focuses on those whose connection to the rest of the group is unusual; and the third on small subgroups that are unusual. Which of these views of significance is relevant depends on the data and the problem domain. However, all three reveal some latent structure in the data that focuses attention onto some nodes and edges.

These approaches rank the nodes (records) by significance. But what about the significance of the edges? In a graph, the edge weights already provide a measure of significance. But graph embeddings also provide an emergent indication of edge significance. In such an embedding, the distance between any pair of points reflects the (dis)similarity between them in a global sense – that is, by integrating the pairwise similarities across the whole graph. Thus, for nodes that are connected, differences between the pairwise edge weight and the distance apart they are in the embedding provides new second-order significance information. Edges are especially significant when they have high local similarity but low global similarity (they are placed far apart); and when they have low local similarity but high global similarity (they are placed close together). Thus computing a suitably scaled difference between the distance implied by local similarity and the distance in the embedding allows edges to be ranked by a new second-order kind of significance.

[1] This is the random walk Laplacian.

A special case is when the local similarity is zero – the nodes are not directly connected. The two nodes concerned still end up at some distance apart in the embedding, so they always have a global similarity. Two unconnected nodes that end up sufficiently close in the global embedding are probably especially significant. For example, this may mean that there is an edge between them that failed to be captured when the data was collected; or that there are multiple *indirect* paths between them, suggesting that perhaps they are concealing their direct connection [11].

An alternative approach to discovering significance is to start from the 'other end' and try to remove information that is either spurious or trivially explainable. One recent approach to this, with special usefulness for data that represents correlation, is random matrix theory. The distribution of eigenvalues of a random matrix, appropriately scaled, is known. Given a data matrix, its eigenvalues can be calculated and the eigenvectors associated with eigenvalues that match those of the known distribution can be discounted. The remaining structure is much more likely to be meaningful [10].

3.2 "Classical" Contextual Modelling

In the metaphorical classical-physics approach to modelling context, the analyst's state becomes a filter on the underlying model, so that the model as presented reflects what the analyst 'already knows'. Knowledge that is already understood may be discounted or de-emphasized, providing scope for heightening attention to the data or parts of the model that are more unexpected, given the current state. However, the underlying model does not change. This avoids several thorny problems: filtering is computationally much cheaper than recomputing the model, and the issue of rollback if the state (that is, the analyst's assumption) turns out to be erroneous is avoided.

Apart from changes in the data, there are three ways in which analyst context can change:

- The analyst may acquire new knowledge from some other source. For example, investigation of an apparently *interesting* record may show that its properties arise from some technical issue, perhaps in its collection. Knowing this, similar records may plausibly be discounted.
- The analyst may have a new hypothesis or opinion, and want to ask a "what if?" question. This new internally generated information, when taken into account, may alter the significance ranking, perhaps revealing some structure that was previously hidden.
- The analyst may have a change in perspective, as the meaning of something already visible in the data or model becomes apparent. Humans are poor at understanding the implications of well-known facts, so analyst state will change by introspection, without any external stimulus, and at unpredictable times as implications of the current state become apparent.

New data is significant. The simplest analyst state is that s/he understood the data and model at some previous point in time ('yesterday'). Significance then becomes a surrogate for 'recent', and the system should emphasize data that has arrived since the previous point in time and the changes in the model that result. Although this is often relatively easy to implement, few systems seem to provide this functionality.

At its lowest level, all data newer than some checkpoint could be labelled as significant. However, when data is plentiful, this may already be overwhelming. A generic way to abstract from large data volume while still detecting when change is occurring is to compute statistics of batches of data, and compare them. Significance is derived from changes in statistics, rather than change in data. The particular statistics used can vary widely, from simple means, medians, and standard deviations of input data to statistics of the predictions across classes, and prediction margins [1, 3].

An important way in which new data becomes significant is if it relates to some previous issue or question with which the analyst was concerned. For example, suppose that an earlier search for a connection between two people did not find one. The arrival of new data that describes such a connection makes the data more than usually significant. If the search took place a long time ago, the analyst may not even remember it, and there may have been many such searches with a negative result, so system support compensates for human weakness well. Of course, this kind of significance can only be detected if the system records analyst activity and matches new data against it. As Jonas has cogently argued, all systems should do this [9].

Analysts look at the data from a different direction. Many analysis tools do provide ways for analysts to alter the view of the data or models to focus on some pieces or perspectives at the expense of others. Tools such as i2 Analyst Notebook or Netmap Analytics [7, 8] allow 'slice and dice' actions to segment data in different ways, map them to timelines, visualize records and/or their connections, including sophisticated rendering using color, multiple simultaneous projections, and fish-eye presentations that magnify detail in one region at the expense of others. So far, the limitations of these tools is that the actions are not inductively generated from the data, but rely on analysts to drive them. Thus they directly support significance ranking for hypothesis-driven investigation; and indirectly support it when analysts can work out and describe which parts of model have altered significance as the result of new knowledge or new perspectives – but this may be difficult, and puts a substantial cognitive burden on the analyst.

Analysts have existing high-level models of significance. Analysts working in an area develop a sense of what kinds of data are likely to be more significant, and working with a particular dataset come to understand some parts that are especially significant or insignificant. These kinds of understandings can be captured by an analyst model. In contrast to an e-learner model, though, such models should be treated conservatively (analysts may think they understand more than they do) and temporary (because adversaries constantly try to exploit analyst blindspots).

If analysts label aspects of the data or models that they think they understand, then it becomes possible to build predictive models that will try to label new data as belonging to the same general class as the understood data (and so less significant) or not (and so more significant). All of the techniques described in the previous section can be reused for this, with "objects I understand" as the 1-class label, rather than "normal objects". In other words, analysts should be able to label some subset of records, or model pieces (e.g. clusters) as understood or especially interesting, and have this information incorporated into the rendering to discount or highlight them, and others like them.

There are several specialized techniques that do not fit well with determining significance (a multiclass problem) but can be useful for modelling analyst context (a two-class problem: 'things I understand' versus 'things I don't understand'). For example, autoassociative neural networks

use standard neural network elements and learn by back-propagation, but have a very small layer in the middle. They are trained to reproduce their inputs on their outputs. The presence of the small layer forces them to do this by learning a compact representation, rather than simply copying inputs to outputs. When an AANN has been trained, the difference between inputs and outputs will be small for any record that resembles those on which it was trained, but large for a record of any other kind.

Dictionary-based compressive techniques also provide a way to quickly determine anomaly of records. Known data is used to train a model that is actually a dictionary, mapping features of records to shorter representations [13]. Once trained, such a dictionary is able to compress records like those from which it was trained well, but other records will be compressed poorly. In contrast to AANNs, a dictionary is a model that can easily be refactored, so that updating to reflect changing analyst knowledge is cheaper.

Using techniques such as these, records that an analyst understands can be used to train a model of this understanding. For example, suppose the model is a clustering and the analyst understands all of the records in one cluster, and a subset of the records in a second cluster. A model trained on the understood records can be used to provide anomaly scores for all of the records, perhaps displayed as an overlaid color code (say, from green to red) on the current model. This should indicate that the understood records have low anomaly, and are colored green. The remaining records in the second cluster might either be also labelled green, in which case the second cluster is internally consistent, or labelled as orange, suggesting that there is important substructure within the cluster. Other clusters will also be labelled with colors ranging from orange to red indicating how anomalous they are.

The standard way in which to incorporate existing knowledge into a model is to use Bayesian techniques, in particular *priors* which encode the inherent probability of certain aspects of the data. Although Bayesian approaches are well-understood and powerful, they may not be of great use in adversarial settings. Prior knowledge is likely to be fragmentary, and its meaning and importance hard to assess. This kind of information is not easy to map to a representation as a probability distribution.

3.3 "Quantum" Contextual Modelling

In adversarial settings, models will often need to be rebuilt to reflect the changing actions of adversaries, and to prevent their discovering weaknesses in models, for example by probing. What I envisage here, though, is much smaller and more frequent recalibration of an existing model to reflect an analyst's (changing) external knowledge or hypotheses. Because such knowledge is not always correct, and because hypotheses are not necessarily correct by definition, it is critical to be able to roll back model changes.

Semisupervised learning. The general field of semisupervised learning is concerned with learning models in settings where there is a large amount of available data, but class-labelled data or known associations among records are rare or expensive [21]. For prediction, a decision boundary is learned from the labelled data, but can exploit the presence of unlabelled data because, intuitively,

boundaries should not pass through regions where records are dense. For clustering, providing knowledge that a pair of records *must* be in the same cluster, or *cannot* be in the same cluster can improve results. Partial information can also be applied to graph-structured data as a regularization that makes it likely that two nodes connected by an edges will have the same label (or be in the same cluster).

Such techniques can be adapted to alter or refine existing models to reflect extra information from an analyst, rather than to build totally new models. For example, the analyst may know that (or want to see what happens if) a particular record should have a different class label. An analyst may know of a connection between two nodes for reasons outside the data itself; or may decide that an apparent connection is spurious and should be removed.

Reweighting. Once data have been normalized, several algorithmic approaches allow extra information to be applied by changing the weights of some parts of the data. For example, the weight of a record, of an attribute, or of a graph edge may be changed – because this is done after normalization the change is a relative one, and plausible magnitudes can be estimated. Upweighting a value makes the record(s) seem more important. For models that consider correlation among data, this has the effect of altering the apparent importance of records that resemble the altered one(s). Thus this process can be used, for records, to "show me more like this", or to increase the impact of an attribute, or to increase the local similarity of a set of nodes in a graph.

Parameter setting. One way to alter a model is to change some of the parameters that were used to construct it. In adversarial settings, model-building techniques with few or no parameters are usually to be preferred, as it is often hard to know what choices of parameters are plausible. If adversaries can guess what parameter choices are likely, they can attempt to manipulate the resulting model. Nevertheless, there are sometimes ways to change parameters to explore ranges of models of the same general kind. For example, the threshold used when mapping pairwise similarities to adjacency matrices can be altered. Because the resulting embedding depends on the entire structure of the graph, changing this threshold is more than changing the rendering of the graph.

Additive versus subtractive analyst knowledge. There are some subtleties in the interaction between analyst and significance computation. If the analyst context adds information to the data, then the altered model will be more conservative than the original model. However, if the analyst context is inherently subtractive, for example by discounting some of the available data, then the model will become less conservative. As an analyst comes to understand, and so discount, more and more of the 'central' structure of the data, there is a tendency to focus increasingly on those parts of the data that are most random. These issues have already surfaced in research aimed at developing curious robots [12] which have to decide autonomously what to do next. Careful exploitation of the fact that significance peaks in the middle of the spectrum of possible forms of new data will help to avoid this pitfall.

Analysts would be helped if systems could point them not only towards areas where significance is high, but also to areas where there is the greatest payoff for understanding, that is areas where significance could most easily be decreased. As far as I know, no work in this direction has yet been attempted.

Overall, what is needed is ways to "quotient" one model by another, so that the most general difference between the two (and its derivatives) could be calculated.

4 Discussion and Conclusions

Computing significance allows knowledge-discovery tools to build models that indicate which parts of their outputs are of greatest importance to a particular analyst in a particular context. Since analysts in adversarial settings are often overwhelmed with data, this provides a way to guide their attention and so improve their productivity.

Significance has both an inductive and deductive component. Inductively, the data and models themselves provide signals about the meaning of new data. I have suggested that it is helpful to categorize these signals into a spectrum: *normal – anomalous – interesting – novel – random*, when deciding their meaning. Deductively, the context of analysts and organizations also signals what data and models mean, suggesting that some aspects are less significant because already understood.

The technical challenge is to incorporate these two components into algorithms that provide significance information as part of model building. Some techniques already include some calculation of significance, although not in a useful way; for example, predictor confidence can indicate data that it *interesting* but not data that is *novel*. New algorithmic techniques are needed to include significance computation explicitly, and to create ways to feed context back into model rendering and recalibration.

References

[1] H. Abdulsalam, D. Skillicorn, and P. Martin. Classification using streaming random forests. *IEEE Transactions on Knowledge and Data Engineering*, 22, 2010.

[2] M. Bourassa and D. Skillicorn. Hardening adversarial prediction with anomaly tracking. In *IEEE Intelligence and Security Informatics 2009*, pages 43–48, 2009.

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

[4] S. Brin, L. Page, R. Motwani, and T.Winograd. The PageRank Citation Ranking: Bringing Order to the Web. *Stanford Digital Libraries Working Paper*, 1998.

[5] K. Bryan and T. Leise. The $25,000,000,000 eigenvector: The linear algebra behind Google. *SIAM Review*, 48(3):569–581, 2006.

[6] E. Davies, G. Gladwell, J. Leydold, and P. Stadler. Discrete nodal domain theorems. *Linear Algebr. Appl.*, 336(51), 2001.

[7] J. Galloway and S. Simoff. Digging in the details: A case study in network data mining. In *Intelligence and Security Informatics, Springer Lecture Notes in Computer Science 3495*, pages 14–26, 2005.

[8] J. Galloway and S. Simoff. Network data mining: Discovering patterns of interaction between attributes. In *Advances in Knowledge Discovery and Data Mining, Springer Lecture Notes in Computer Science 3918*, pages 410–414, 2006.

[9] J. Jonas and L. Sokol. *Data Finds Data*, chapter 9. O'Reilly Media, 2009.

[10] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud. *Random Matrix Theory and Financial Correlations*. World Scientific, 1999.

[11] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 556–559, 2003.

[12] J. Schmidhuber. Developmental robotics, optimal artifical curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[13] J. Schmidhuber. Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. In V. Corruble, M. Takeda, and E. Suzuki, editors, *Proc. 10th Intl. Conf. on Discovery Science (DS 2007)*, LNAI 4755, pages 26–38, 2007.

[14] D. Skillicorn. Detecting anomalies in graphs. In *2007 IEEE International Conference on Intelligence and Security Informatics*, pages 209–216, 2007.

[15] D. Skillicorn. *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. CRC Press, 2007.

[16] D. Tax. *One Class Classification*. PhD thesis, Technical University Delft, 2000.

[17] G. Treverton. *Reshaping National Intelligence for an Age of Information*. Cambridge University Press, 2001.

[18] G. Treverton. Risks and riddles. *Smithsonian Magazine*, June 2007.

[19] A. Türker, I. Görgün, and O. Conlan. The challenge of content creation to facilitate personalized elearning experiences. *International Journal on E-Learning*, 5(1):11–17, 2006.

[20] U. von Luxburg. A tutorial on spectral clustering. Technical Report 149, Max Plank Institute for Biological Cybernetics, August 2006.

[21] X. Zhu and A. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artifical Intelligence and Machine Learning 6. Morgan & Claypool, 2009.