Inquisitive Pattern Recognition

Amy L. Magnus
Intelligent Information AFRL/IFTD
Rome Research Site
Rome, NY, U.S.A.
amy.magnus@rl.af.mil

Steven C. Gustafson
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright Patterson AFB, OH, U.S.A.
steven.gustafson@afi.t.af.mil

Abstract - In nature, inquisitiveness is the drive to question, to seek a deeper understanding, and to challenge assumptions. Within the discrete world of computers, inquisitive pattern recognition is the constructive investigation and exploitation of conflict in information.

Data fusion is fertile proving ground for inquisitive technologies. Multi-source, multi-modal data inherently contain conflicting information. As data fusion advances capabilities in situation assessment, strategies to identify and resolve conflict become important.

Inquisitive pattern recognition (IPR) is a persistent, unsupervised learning capability whose concepts include falsification—similar to the supervised learning technique of cross validation—and the classification of confusion in feature space. Coupled with knowledge base technologies, inquisitive pattern recognition enables a computer to acquire new experiences.

Keywords: Unsupervised learning, cross validation, persistent learning, falsification, relevancy, pattern recognition, knowledge base.

1 Introduction

For data fusion, there are benefits in understanding that confusion may as likely result from bad alignment models as from bad data. Noise must be discarded, but data that are merely poorly aligned are still of use.

In this paper, we present inquisitive pattern recognition—a capability for persistent unsupervised learning. We apply this methodology to data fusion—modeling multi-source information in a way that facilitates the identification and classification of conflict between sources. Such methods culminate in the recognition of misaligned data and also value-added information from single sources.

1.1 Uniqueness in data fusion

From an information management point of view, data fusion is beneficial in two ways: (1) It improves signal to noise ratios, and (2) it increases the information bandwith of a multi-source system—allowing more unique information to be captured and processed simultaneously. [1]

Basically, data fusion aligns redundant information from disparate sources and then further refines this information through the incision of source data that is both unique and relevant.

Aligning redundant information improves signal to noise ratios[2], and the assimilation of unique data increases the information bandwidth of a data fusion system[3]. To realize both benefits, data fusion algorithms should include three basic processes: (1) the alignment of redundant data, (2) the assessment of unique information, and (3) the assimilation of all relevant information—unique and redundant—into an exhaustive world model [2,4,5].

Multi-source systems with overlapping mono-modal sensors focus on the alignment process—reducing signal to noise ratios to maintain high confidence in target detection. In systems with multi-modal sensors or sensors with little overlap, assessment becomes the dominant process—siphoning value-added data from each sensor to form an understanding of target behavior. [4]

1See Figure 1 and Table 1.
Pattern recognition (PR) is the process of discovering and associating patterns in reference to a task[6]. Classical pattern recognition is concerned—rightly—with representing knowledge fully and accurately in the computer, and traditional methods investigate trends and consistencies in data.

Contrarily, inquisitive pattern recognition investigates the inconsistencies. Further, it seeks to assign task-related meaning to the biased or volatile portions of a knowledge representation. The results of this investigation are to be used by a computer program to adapt its internal knowledge representation and to conform to an altered task.

In data fusion, inquisitive pattern recognition examines conflicting observations from disparate sources in reference to the natural order of data collection. Such an examination tenders clues over what is real, but generally unanticipated.

2 Inquisitive Pattern Recognition

The focus of inquisitive pattern recognition is to formulate a mathematical basis for discovering confusion—those unstable regions in information modeling that lead to poor decision making. Further, these mathematical treatments are indexed to task-savvy knowledge bases with the goal of inferring which unstable regions in feature space hold the most promise for yielding important knowledge discoveries.

The key components of inquisitive pattern recognition are falsification, confusion recognition, and relevancy testing.

- Falsification identifies the regions of feature space where decision making is known to be poor.
- Confusion recognition connects decision-related meaning to these regions in order to direct conflict resolution processes.
- Relevancy testing promotes important discoveries.

These components are presented in the next three sections and supported by a simple illustration of an object recognition problem.

2.1 falsification

Computers only know what we tell them. Computers only know, they don’t doubt. This arrogance requires users to conform to a program’s way of doing things.

Doubt is, in fact, the first step in unsupervised learning; it initiates the process. Cognitive science teaches us that humans predict through a set of beliefs, and we learn by first refuting present beliefs[7].

If a computer “doubts” itself, this means it has the capability of falsification—the ability to autonomously recognize failings within its programs’ own knowledge representations. Consequent learning benefits from having a focused goal and a definable context[8].
Falsification provides discretion. We envision persistent learning as a reaction to a stimulus, not a full-on full-time process. Instead, falsification is intended to identify learning opportunities, concentrating on those portions of a program’s knowledge representation in most need of attention.

Similar to cross validation, falsification gauges the effective complexity of a knowledge model. Complexity conveys the ability of a knowledge representation to capture the underlying data structure of task-related observations. The worthiness of a knowledge model is completely determined by its competency in analyzing new data[9,10]. Cross validation operates in the initial design of a model to predict the complexity of a given task[10]. Falsification works in the model’s application to empirically measure task complexity.

Cross validation is used in supervised learning to select an appropriate stopping point for training. This process separates design exemplars into two sets: a training set and an evaluation set. A knowledge model is interpolated from the training set while the evaluation set is held back. An error trend is drawn from the evaluation set and, when this trend begins to increase, training stops. [10]

Cross validation stops training before memorization occurs—halting the process when an appropriate level of complexity in the representation is achieved. Error can be tracked because learning is supervised, but error trends for the training set tend to be misleading. Given a convergent learning algorithm, the error from the training set will continue to decrease until that set is memorized. Memorization is undesirable; it defeats the purpose of training since the resultant knowledge representation does not generalize well in recognizing new data. [10]

Falsification differs from cross validation in that it is not used to select an appropriate representational complexity but to test the design-specified complexity. Again, cross validation picks the best complexity—or rather the best guess using what labeled data is available at design. In application, falsification tests this best guess against trends implied by unlabeled observations.

Falsification requires the separation of prior knowledge into multiple prior opinions, i.e. generalizations of the parent knowledge model, or the comprehensive opinion. The comprehensive opinion is discredited where prior opinions vary significantly. Contlicting prior opinions apprehend weaknesses in a knowledge model—weaknesses that must be addressed through augmentation, either additional prior knowledge or supplementary observations or both.

Falsification algorithms are knowingly naive—containing incomplete belief structures that break conspicuously so that the computer can examine their fallacy in action. Falsification is powerful because, though in unsupervised learning observations are not labeled, the relationships between prior opinions can be specified.

Falsification can also be used to test beliefs, or logical groupings within a knowledge model. As shown in Figure 2, new observations are collected and considered in the context of each prior opinion, forming respective event hypotheses.

If hypotheses for an event vary significantly, we can reason that either certain beliefs are not appropriate to present observations or they are obsolete in light of recent evidence. Ideally, we want to be able to localize and replace inadequate beliefs—customizing knowledge without losing what still works.

Falsification algorithms are knowingly naive—containing incomplete belief structures that break conspicuously so that the computer can examine their fallacy in action. Falsification is powerful because, though in unsupervised learning observations are not labeled, the relationships between prior opinions can be specified.

Falsification requires (1) disparate prior opinions and (2) a measure quantifying significant confusion among either prior opinions (for complexity testing) or event hypotheses (for belief testing).

Here is a simple illustration of an object reconstruction problem. For this illustration, prior opinions are contrasted in the falsification process.³ Consider the triangle of Figure 3. The triangle is a real-world object that cannot—for the purposes of this illustration—be represented completely. To reconstruct the triangle, we are given two separate basis sets—a set of

³ Persistent learning is the learning that continues after a knowledge representation transitions from design to application. See Section 3 for further detail.

³ Section 3.2 gives an example of falsification where event hypotheses are contrasted.
circles and a set of squares—to fill in the two-dimensional space.

Figure 3: The target object and two representational sets—one comprised of circles, the other of squares.

The two basis sets are used to form two disparate prior opinions. For this example, we’re restricted to using only one basis set at a time. This is a realistic limitation paralleling restrictions in imaging methods used to capture real world objects. In a simplistic way, the circles correspond to imaging onto film and the squares correspond to scanning the object and storing an image matrix.

Two representations of the triangle are formed by filling the two-dimensional space—first with the circle basis set, then with the square basis set. An OR is performed to merge the space filled by the circles. The same operation is applied separately to the squares.

Due to the limitations of the basis sets, neither the circle-based prior opinion nor the square-based prior opinion captures the entire essence of the triangle. The missing essence, as seen in Figure 5, we refer to as the error space. For this illustration, we form each error space by applying an XOR operation to the triangle and each prior opinion in turn.

Often, we don’t have perfect knowledge of the real world object; so, normally, we can’t calculate the error space directly. Instead, the best we can do is evaluate the confusion between opinions.

Figure 5: The two prior opinions and their respective error spaces. A n XOR operation is applied to the triangle and each prior opinion in turn to form the error spaces.

Here is where we perform falsification. The prior opinions are contrasted—excluding the spaces they share—to form the confusion space shown in Figure 6. The confusion space is where contrapositive logic allows us to state with certainty that the object is not well understood.

Figure 6: The confusion space—formed by applying an XOR to the prior opinions.

Again, falsification identifies the regions of feature space where decision making is known to be poor.

Confusion is not error. Nor is it a complete measure of uncertainty, as confusion space does not include spaces where it is likely that decision making is poor. But it is a good starting point, and with a little more manipulation, a better estimate of the error space and thus a better understanding of the original object can be formed.

The tools for this additional manipulation are confusion recognition and relevancy testing. Respectively, these persistent learning skills partition the confusion space and prioritize the refinement of the partitions.

Confusion recognition interprets features in confusion space by leveraging the known idiosyncrasies of the applied basis sets. The recognition process divides confusion space and associates meaning with each piece.

---

See Figures 4 and 5.
“When you have eliminated the impossible, whatever remains, however improbable, must be the truth.”
Sir Arthur Conan Doyle
The Sign of Four [1890]

Inquisitive pattern recognition that stops at falsification is rather weak. Doubt may initiate the learning process but, if the cause of confusion is not known, then context is unclear and learning unregulated. As a result, the conclusions that can be drawn from mere falsification are severely limited and often regressive[14].

When forming new experiences, the cause of confusion is decidedly not known[8]. Instead, one might go through the process—as the fictional Sherlock Holmes does in The Sign of Four—of eliminating potential causes.

Take, for example, the three-class problem of separating apples from oranges from tangerines. If we can eliminate the choice of—say—apples as contributing to the confusion of one region, we can narrow down the context to a choice between oranges and tangerines and bound the learning opportunity that region constitutes.

The true power in confusion recognition is not found in associating task-related meaning, as in our apples-to-oranges-to-tangerines example, but in associating decision-related meaning to confusion space.

In the modeling of decision boundaries, confusion occurs in transitional regions between what is believed to be true and what is believed false[12,13]. Transitional regions do not all look alike and (This is key!) are as much a function of the basis sets we choose to model a decision as the form of the decision itself. If this were not true, the confusion space of Figure 6 would be null space.

Confusion recognition separates confusion space into distinct regions that may benefit from being resolved in different ways.

For instance, in Figure 7 we’ve divided regions of confusion space into two categories—corners and borderlines—and assert that the confusion associated with corners is significantly different from the confusion associated with borderlines. From this assertion, we can infer that any given strategy to resolve the confusion within these disparate regions will yield disparate results.

Confusion recognition enables the capture of new experiences. It serves to bound unsupervised learning by associating meaning to new information that is either task-related or—more generally—decision-related. When learning is properly bounded, it is more efficient as training requires fewer exemplars and fewer features. This lessens demands placed on the collection of new training data. Moreover, fewer features enhance the probability of reducing confusion.[9,10]

Figure 7: Confusion recognition. The confusion space of Figure 6 can be partitioned into six separate regions. These regions have then been classified into two categories, corners and borderlines.

The last and crucial step of inquisitive pattern recognition is relevancy testing—assigning priority to learning opportunities. Before attempting to resolve confusion, it is best to weigh the benefits and risks posed by each opportunity—pursuing only those where there’s both sufficient evidence and justifiable need. In the end, relevancy testing determines what, if any, action should be taken to refine a knowledge model.

Relevancy testing calls for an excellent understanding of how the limitations of the applied basis sets hinder the goals of the computer task. Again, take our object reconstruction illustration: Note that the corners of the triangle are not engulfed by the confusion space but that the borderlines essentially are. Here it is more important to determine that the corners are poorly modeled. This declaration requires not only an understanding of representational shortcomings—acute corners are difficult to fill—but also an appreciation for the task at hand—object reconstruction.

Relevancy testing must cross-index representational limitations with task goals, and this is where our effort to build task-savvy, decision-oriented knowledge bases comes in. Within these knowledge bases, we are endeavoring to capture the idiosyncrasies of our favorite representational tools—tools that include data pyramids, wavelets, and neural networks. These knowledge bases provide the reasoning resources (1) to cross-index specific task-related knowledge with decision-making expertise and (2) to infer from this mapping which learning opportunities constitute important discoveries.
Falsification triggers learning when the refutation of one belief can be combined with other knowledge to further the task at hand[14].

Inquisitive pattern recognition seeks to harness the power of falsification by augmenting the process with confusion recognition and relevancy testing. These additional skills provide a means to form new experiences by bounding and regulating unsupervised learning opportunities.

Supporting research involves the generation of knowledge bases that grasp the mechanics of mathematical representations and capture inherent implications to decision making.

Inquisitiveness is part of an overall strategy to design persistent learning into computers—in large part, to delegate to them the maintenance of knowledge representations.

Persistent learning is the learning that continues after a knowledge representation transitions from design to application. Initial design-stage learning creates a generalized knowledge model in a canned, offline environment. Persistent learning customizes the model online—“on the fly”—using real world observations.

Learning is active, cumulative and incremental. For persistence to be viable, learning must also be stable. Successful inquisitive pattern recognition regulates the incremental, cumulative adaptations of a knowledge model and ensures the principal computer task remains viable.

Our model for persistent learning—chosen for its completeness—is the Prochaska/DiClemente change wheel[15] shown in Figure 8. The change wheel—first developed to model human behavior modification—is a multi-stage feedback process. Persistent learning has seven stages: steady state, falsification, discovery, determination, action, maintenance, and relapse.

Principle task:
• Steady state is the principal computer task; it accomplishes prediction and dictates the pace of learning.

Passive learning stages:
• Falsification initiates learning. Upon a falsification stimulus—a contradiction to present beliefs—the change wheel is entered.
• Discovery is an investigative process. Tasks include the collection and evaluation of evidence that supports modifying the program’s knowledge model.
• Determination is a decision process. Here, recommendations are made regarding appropriate modifications to present beliefs.

Active learning stages:
• Action is the actual modification process where the knowledge model is altered.
• Maintenance involves various sustainment actions. Tasks include minor adjustments to the knowledge model, setting exit criteria for permanent exit from the wheel, and setting relapse criteria.
• Relapse handles the termination of an intervention cycle in preparation for a new cycle. Tasks may include archiving of data from the completed cycle and may also involve a partial or complete regression to the former knowledge model.

Data fusion is a type of persistent learning process where a world model—a unified representation of multi-source information[2]—is periodically updated. Inquisitive pattern recognition can help answer two key questions in the maintenance of a world model: (1) what new observations should be incorporated and (2), more generally, what kind of information needs to be portrayed in the world model.

Figure 9 matches persistent learning skills to the three basic processes of data fusion algorithms presented in Section 1.1. To summarize:
• Alignment[^5] is the steady state task and is not associated with the learning stages as it leverages redundant information—information that should be well understood.

---

[^5]: Alignment affixes data from disparate sensors to a common reference.
• Assessment\(^6\) corresponds to the passive stages of the wheel.
• Assimilation\(^7\) encompasses the active stages.

Again, inquisitive pattern recognition’s contribution is to the assessment of unique information: Where is the value-added information, and what opportunities does this information offer? In Section 3.3, we apply our methods to answer these questions in an image fusion application.

Using the IPR methodology, we have designed an image fusion algorithm based on super resolution. Super resolution is an object reconstruction technique that fuses a series of offset low-resolution images to reconstitute the high frequency content of a real-world object\([16]\).

Inquisitive super resolution addresses shortcomings in other super resolution methods. There is a tendency in non-linear methods to over-smooth edges and periodic patterns within the theoretical reach of the reconstructed resolution\([17]\). Over-smoothing occurs due to the application of an imperfect assumption—pixel values are unaliased. This presumption is awkward considering the application.

In the super resolution algorithm developed by Cheeseman et al at NASA\([17]\), pixel error is assumed have a monomodal Gaussian distribution. This is a fair assumption for slowly varying images\([18]\). However, if there’s detail in the image, multiple random processes with different means corrupt pixel values. Given this is the case, conditions for the central limit theorem\([19]\) are not met, and the monomodal assumption breaks down frequently.

Inquisitive super resolution (ISR) addresses the breakdown of pixel error assumptions by testing the validity of these assumptions pixel by pixel. Figure 10 gives a pictorial summary of the algorithm, and pseudo code follows.

\(^6\) Assessment evaluates aligned data to identify hidden information and identifies conflict in the world model.
\(^7\) Assimilation modifies the world model melding together aligned data and assessed information.
% Inquisitive super resolution
% Get observations and initiate estimates
store Y: set of offset, undersampled images
initiate V: set of pixel states
initiate Z: fused object estimate
% Initiate pixel set for cross validation
select Z.detail: subset of “interesting” pixel values
% Optimization loop
t=0;
epochError = 1;
validateError[0] = sum(abs(Z.detail));
while (epochError > 0) {
  t++
  % The 3 core fusion steps
  estimate S: set of registration parameters
  update V
  update Z
  validateError[t] = sum(abs(Z.detail[t-1]-Z.detail[t]));
  epochError = validateError[t] - validateError[t-1];
}

The core fusion steps in inquisitive super resolution are:
1. Alignment—the estimation of registration parameters S which reference the offset images of Y to the object Z;
2. An assessment—the estimation of pixel states V; and
3. A simplification—the estimation of object Z.

Not included in the NASA algorithm[17], the assessment of pixel states is an added step which calls for falsification, confusion recognition, and relevancy testing:
• First, falsification highlights anomalies. Once the input images are captured, interlocking interpolations of each image are calculated to test the local area of each pixel for high frequency content.
• Next, confusion recognition classifies the state of each pixel in Y—whether it is well behaved, aliased or noisy.
• Finally, relevancy testing weighs the apparent severity of aliasing—gauging the likelihood of converging toward a reasonable pixel estimate given the available evidence.

The iSR algorithm responds to the quality of data it is given. This flexibility benefits resource management as well as object estimation.

Inquisitive pattern recognition is a persistent unsupervised learning capability. It harnesses the power of deductive analysis within a computer program by augmenting the falsification process with confusion recognition and relevancy testing. These skills provide a means to bound and regulate unsupervised learning.

Currently, we are applying this methodology to image compression, machine vision, and non-linear image fusion applications such as the inquisitive super resolution algorithm depicted in Figure 10. Also in progress are general interpolation techniques coupled with unsupervised evaluation tools. For example, we are conducting research into neural network designs that facilitate rule extraction from the mathematical representation. Contributions anticipated from this research are falsification metrics that test the validity of an extracted rule in application.

Inquisitive pattern recognition is a step toward the ultimate goal of machine learning—where a computer identifies a learning problem, defines a self-supervised experiment, interprets the results of that experiment, and finally applies these results in a practical manner.