Using Qualitative Domain Proportionalities for Learning Mission Safety in Airspace Operations

Geoffrey Levine[1] and **Ugur Kuter** [2] and **Kevin Van Sloten**[3] and **Gerald F. DeJong**[1]

Derek Green[4] and **Antons Rebguns**[4] and **Diana Spears**[4]

Abstract

This paper focusses on an automated learning/reasoning system for inferring mission safety in airspace operations. We describe a simplified version of a realistic airspace operations scenario that inspired our work. Our domain knowledge about airspace operations and mission safety is expressed qualitatively. We define and describe a way to construct explanations of missions that are translated into a Neural Network representation which is fit to the data and scored. We describe a pruning algorithm to select a greedy-best explanation structure of mission safety. Our experimental evaluation demonstrates the effectiveness of using domain knowledge in learning compared to the standard hidden-layered Artificial Neural Networks.

1 Problem and Significance

One of the most important factors in military decision-making is *safety*, i.e., whether a high-valued asset participating in a mission is safe or not, given the enemy threats and our capabilities in that mission. For example, consider the following *fictitious* but very realistic scenario, simplified substantially here:

> *Intelligence confirms that the militant power in the recently captured Area* 6 *have escaped to neighboring countries such as Areas* 1 *and* 7 *with large numbers of weapons, including SCUD missiles and hand held surface-to-air missiles (SAMs). Despite this intelligence, the exact location of these missiles is still unknown. The Joint Force Air Component Commander (JFACC) directs a large proportion of airpower from Areas* 3*,* 5*, and* 6 *to be devoted to finding and eliminating the threat of SCUD missiles. Assets involved in this mission include F-15E Strike Eagles, F/A-18 Hornets, JSTARS, E-2C Hawkeye, E-3A AWACS, and KC-135 tankers. The Rules of Engagement (ROE) require that (1) the coalition forces are not authorized to engage or otherwise cross the border into any neighboring country, and (2) a visual confirmation of any SCUD launched is established before any action – thus, a SCUD hunting mission will only occur during daylight hours or under three-quarter or more moon clear nights.*
>
> *The air defenses of the potential host countries that would interfere with any kind of SCUD hunting mission include multiple shoulder launched surface-to-air missiles, along with the TOR and Hawk (export variant) missile systems. Figure 1 shows some Hawk and TOR missile ranges, and some suspected launcher sights depicted by yellow stars. Hawk missile sights are shown in red, and TOR in orange.*
>
> *Intelligence indicates a SCUD missile will move out of hiding to a position in Area* 7 *– to launch a chemical armed weapon against a concentration of friendly coalition forces in Areas* 3*,* 5*,, or* 6*. The SCUD hiding site is*

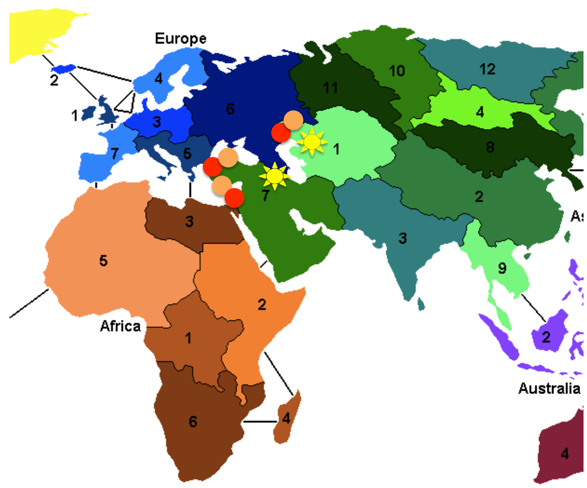

Figure 1: An illustration of the military planning scenario described in the text. The map is based on a screenshot of the map of the well-known board game Risk [Wikipedia, 2009].

[1]University of Illinois at Urban, Computer Science Department, Siebel Center for Computer Science, 201 N Goodwin Ave, Urbana, IL 61801-2302, USA. Email: {levine,mrebl}@cs.uiuc.edu

[2]University of Maryland, Institute for Advanced Computer Studies, College Park, MD 20742, USA. Email: ukuter@cs.umd.edu

[3]Analytic Services Inc., 2900 South Quincy St., Arlington, Va 22206, USA. Email: Kevin.VanSloten@anser.org

[4]University of Wyoming, Department of Computer Science, 1000 E. University Avenue, Laramie, WY 82071, USA. {anton,derekg,dspears}@cs.uwyo.edu

approximately 15 miles inside Area 7 from its northern border, with a paved road to the border crossing and numerous unimproved roads that cross the border in to Area 6. The Joint Force Commander has directed the JFACC to find and destroy the SCUD launcher once it crosses the border.

This scenario demonstrates that typically there are a large number of possible factors that affect the decisions and plans for carrying out a military operation safely. For example, an airspace planner (e.g., military commander) must take into account many factors, such as the types of assets (i.e., aircraft) in the friendly and enemy airspace forces, their possible configurations and orientations, the rules of engagement for the mission, weather conditions, available logistic support, resource constraints, operational and tactical objectives, mission priorities, and so on. The space of possible plans is typically enormous, whereas the subset of plans leading to safe missions is typically small.

2 Overview

This paper describes our novel AI tool for helping human military planners to manage the complexity in developing safe mission plans. The basis of our tool is to represent general background knowledge regarding military operations as a set of qualitative proportionalities (e.g., relationships that describe whether increasing the value of one parameter increases or decreases the value of another parameter). Such knowledge is important for an AI system to even have a chance of deployment because (1) it enables the system to learn from a few examples; (2) it provides guidance to a learner in an otherwise very complicated domain; and (3) perhaps, most importantly, it ensures that anything learned by the system conforms with the commanders' decisions, military practices and doctrines.

Our contributions are as follows:

- We present a formalism based on qualitative representations [Kuipers, 1994; Forbus, 1997] to capture expert background knowledge about a domain. We describe this formalism in the context of airspace planning. We describe how to map a qualitative representation of a domain into an explanation structure for a given concept in the domain.
- We describe a way to translate the qualitative representation of background knowledge and corresponding explanation structures into Artificial Neural Networks (ANNs) [Bishop, 2005]. Once an ANN-based model has been initialized with the domain knowledge, it is updated using training examples (scenarios labeled with the degree to which they are "safe"). Since we built ANNs via the given background knowledge, training of them requires few examples only, much less than the number of training examples required for a typical ANN-based learning technique.
- Our learning system performs training (i.e., learning) by using standard gradient-descent algorithms. Learning from the training examples converts the initially abstract qualitative relationships to concrete quantitative relationships.
- We present the results of an experimental evaluation of our approach on a set of reconnaissance missions in the military context provided above. To evaluate the system, the model's predictive accuracy is tested on an independent (from the training examples), previously unseen set of testing examples. We describe our experimental results and conclude with a summary and directions for future research.

3 Qualitative Airspace Planning Knowledge

Compared to logical representations, qualitative representations [Kuipers, 1994; Forbus, 1997] can be more convenient and natural to capture an expert's domain knowledge. Our qualitative assertions represent increasing ($\xrightarrow{+}$) and decreasing ($\xrightarrow{-}$) relationships between entities; $X \xrightarrow{+} Y$ means that, other things being equal, the dependent variable Y will increase (decrease) as the independent variable X increases (decreases), while $X \xrightarrow{-} Y$ denotes the reverse: that Y will generally decrease (increase) in response to X increasing (decreasing). We call X above the *antecedent* of the assertion and Y the *conclusion*. As an example, the assertion $SafetyFromAircraft \xrightarrow{+} MissionSafety$ states that greater safety from enemy aircraft threats generally results in a safer mission; here, the antecedent is $SafetyFromAircraft$, while $MissionSafety$ is the conclusion.

The domain knowledge introduces unobservable or latent variables that the expert finds useful to capture underlying regularities. Our componential domain knowledge allows larger structures to be composed by inference (i.e., chaining through domain knowledge statements). An inferred structure is *well formed* if latent variables occur only internally, with the structure's antecedents all observable, and with the structure's conclusion a class label to be assigned.

Any well-formed inferential structure that is consistent with some observed quantitative behavior is an *explanation* for that observed behavior. Just as the same outward behavior can be manifested by quite different internal mechanisms, incompatible explanations can be constructed from the same observations.

Although our system is *initialized* with qualitative knowledge, learning from the training examples leads to *quantitative* instantiation and refinement of the initial abstract knowledge. This methodology of asking the domain expert to provide qualitative rules, along with labeled examples, from which quantitative rules are inferred, is novel and we have found it to be highly effective. In addition, our domain expert found it to be a considerably more straightforward process than one in which he would have had to provide quantitative rules at the outset.

4 The Domain Theory for Airspace Operations

Our domain knowledge has been simplified for the sake of tractability. Notably, it omits the Rules of Engagement and weather. Our subject matter expert, coauthor Kevin Van

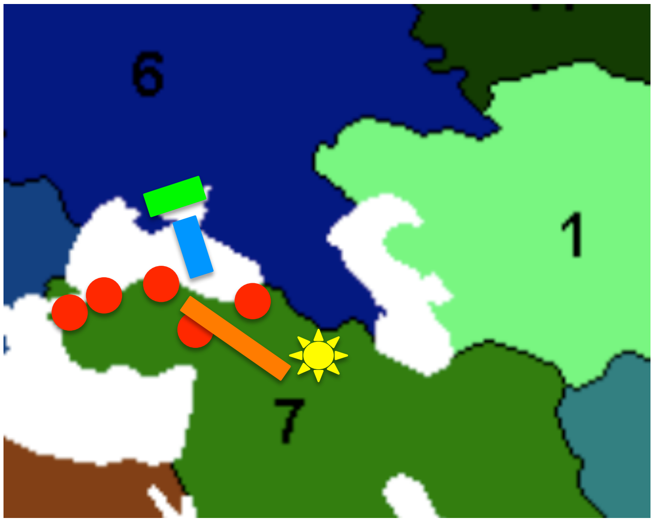

Figure 2: An illustration of an airspace protection mission. Above, all airspace assets and threats are shown as bars. Above, the Green is a high-value asset, Orange is an enemy air threat, Red are enemy missile launchers and their ranges, and Blue is protecting Green from Orange.

Sloten who is an expert airspace manager, provided realistic but manageable scenarios to test our proof of concept system. We do not believe these omissions would require further conceptual advances.

Here we discuss a few domain knowledge expressions to give the reader an idea of the type of knowledge employed. The knowledge includes a list of features, qualitative domain proportionalities (relationships) between the features, facts from a particular scenario such as distances and angles between entities (e.g., aircraft or missiles), mission-related knowledge, and levels for threats, priorities, and so on.

Consider the mission scenario described earlier:

> *Intelligence indicates a SCUD missile will move out of hiding to a position in Area* 7 *– to launch a chemical armed weapon against a concentration of friendly coalition forces in Areas* 3*,* 5*,, or* 6*. The SCUD hiding site is approximately 15 miles inside Area* 7 *from its northern border, with a paved road to the border crossing and numerous unimproved roads that cross the border in to Area* 6*. The Joint Force Commander has directed the JFACC to find and destroy the SCUD launcher once it crosses the border.*

Figure 2 illustrates a potential mission that JFACC could undertake in this scenario. In the presentation that follows, the colors and shapes refer to entities in Figure 2. A blue bar represents an airspace region for the combat air patrol (CAP) of protective friendly fighters, a green bar represents the region of the high-valued asset (HVA) which is to be protected, a red circle depicts the range (based on intelligence) of a surface-to-air missile, and an orange bar is the best estimate (also based on intelligence) of the location of one or more enemy aircraft patrols.

- The mission objective is to perform reconnaissance in the context of SCUD missile hunting under five distinct external *Threat Levels*: low, low-med, medium, med-high, and high.
- As the general level of threat increases, the mission safety decreases: $ThreatLevel \xrightarrow{-} MissionSafety.$
- There are three different levels of mission *Priority* (low, medium, high); the same safety value is less acceptable the higher the mission priority: $Priority \xrightarrow{-} MissionSafety$
- Every high-valued asset (HVA) must have a Combat Air Patrol (CAP) assigned to protect it. For example, in Figure 2, Blue's mission is to protect Green from the infiltrator Orange (which is protected by the Red missile ranges).
- Aircraft always fly within a bar region in an orbit, i.e., they fly back and forth. The length and angle of the rectangular bar indicates the direction and distance of movement. Their orientation is important for sensing.
- The angle from a Blue bar's axis to a threat is measured as the cosine of its nose-off angle to the threat computed as a dot product of unit vectors. Increasing this measure, increases the positional protection ability. For example, for each surface-to-air missile site: $AngleCAPtoMissileThreat \xrightarrow{+} PositionSafetyfromMissile$
- Fighters fly in pairs: the head and the wingman. Thus, there are either two or four fighters are in a blue bar. Similarly, either two or four enemy aircraft are in an orange bar.
- Each fighter comes as a well-defined package: i.e., its capabilities (i.e., its maneuverability, the amount and kinds of weapons it can carry) depends on the aircraft type. Enemy aircraft are always of type Mig-29, but blue (friendly) aircraft can be of type FA16 (Fighting Falcon), F15, or FA18 (Hornet). These are ordered by increasing capability so that the adequacy of an aircraft's weapons is positively influenced by its type: $CAPAircraftType \xrightarrow{+} AircraftWeaponAdquacy$

The most complete explanation possible for Figure 2 using our domain knowledge is shown in Figure 3. As we shall see, this is not necessarily the best explanation for the training observations, but it demonstrates the structure of an explanation. Observable features compose the leaves or antecedents of the structure. The conclusion is $MissionSafety$ which is to be assigned by the classifier. Internal nodes are latent variables introduced by the expert to capture domain distinctions.

5 Learning How to Predict Mission Safety From Qualitative Proportionalities

This section describes a novel Artificial Intelligence tool to represent and learn how to make predictions about the safety of airspace missions, given a qualitative domain theory and a number of past experiences on such missions. We give by our definitions and formalism, briefly, and afterwards, discuss our approach in detail.

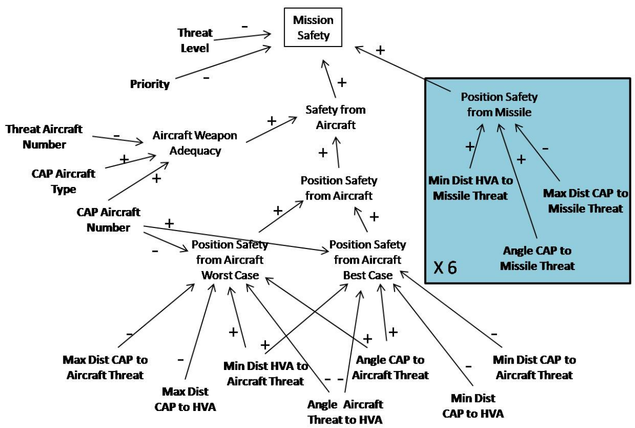

Figure 3: An example of an explanation structure for the airspace-protection mission depicted in Figure 2.

5.1 Preliminaries

Let F be a set of features that describes airspace operations. We described some of the possible features for our scenarios in the previous section; depending on the particular class of missions, F may contain other features that were not listed above. Each feature $f \in F$ has a set of *values* that f can take from.

We define a *mission scenario* as a set of feature-value pairs over a fixed subset of F, known as the *native features* of the domain.

Let D be the qualitative domain theory given for a class of airspace operations. We formalize an *explanation* as a directed acyclic network DAG $X = (V, R)$. V, a subset of F, is a set of features that appear in the explanation.

Each edge in R of the DAG describe a qualitative relationship from D between two features in V: i.e., R is a set of qualitative relationships of either of the following forms: $f \xrightarrow{+} f'$ or $f \xrightarrow{-} f'$).

We deem an explanation $X = (V, R)$ *valid* if all non-native features in V are the conclusion in at least one qualitative assertion in D, and the output node (safety in our case) is reachable from all features in V. In other words, every node in an explanation graph is relevant to the output node, safety. Figure 3 shows an example of an explanation structure.

5.2 From Explanations to Neural Networks

Given an explanation X based on qualitative domain theory D, we define an *Artificial Neural Network (ANN)* [Bishop, 2005] representation of X, denoted as $\nu(X)$, as follows. A neural network unit is constructed for each feature in V. Then for each qualitative assertion $f \xrightarrow{+} f'$ or $f \xrightarrow{-} f'$ in X, we add a weighted directed edge in the neural network structure from f to f'. Let $w(f, f')$ be the weight associated with this edge. Assuming X is valid, our derived neural network structure is a feed forward network, in which all non-native features have one or more incoming edges.

Mathematically, $\nu(X)$ operates as follows. Given a scenario, m, all native feature nodes are set to the value of the corresponding feature in m. Non-native features are defined using a sigmoid function:

$$f = \left(1 + e^{-o(f) - \sum_{f' \in pred(f)} w(f', f)}\right)^{-1},$$

where $pred(f)$ is defined as the set of features, f', for which there is an edge $f' \rightarrow f$ in X. $o(f)$ is a feature specific offset weight.

We use the ANN representation of explanations in order to learn the definition of "mission safety" as a function of the other features that appear/affect in a mission scenario. For this, an important definition is that of the *error function*, which is a measure of how far away we are from an optimal solution to the problem that we want to solve.

As the error function C in our ANNs, we use the standard expected squared-error defined as follows:

$$C = E[(safety(m) - y)^2],$$

where $safety$ denotes the neural network "mission safety" output, m is a mission example, and y is true safety of that mission. Above, E denotes the expectation over the squared error. Practically, let M be the set of training sample mission that are given. Then we approximate the above formula as

$$C = \frac{1}{|M|} \sum_{m \in |M|} (safety(m) - y)^2,$$

where $|M|$ denotes the number of missions in M.

The parameters of $\nu(X)$ are the feature offset weights, $o(f)$, and the edge weights, $w(f, f')$. The learning procedure initializes the ANN representation of X using the following weight function. For each edge in X: if an edge from f to f' in X was created due to a qualitative assertion $f \xrightarrow{+} f'$ then $w(f, f') = 1$; otherwise, if it created due to an assertion $f \xrightarrow{-} f'$ then $w(f, f') = -1$. These weight settings ensure that the initialized neural network adheres to the qualitative assertions. Offset weights are initialized such that the expected output of each sigmoid unit is .5.

Given a set of training scenarios, the learning procedure uses the standard back-propagation technique based on gradient descent [Bishop, 2005] to update edge and offset weights. In summary, a gradient-descent algorithm iterates through the training examples, propagating the errors backwards from the objective (i.e., the Mission Safety in our case) nodes to the inner nodes. This back-propagation is used to compute the gradient of the error, which is then used to guide an update to the parameters. Back-propagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited [Bishop, 2005]. In addition, the technique of "early stopping" [Sarle, 1995] is used to avoid overfitting the data. Details appear in the evaluation section.

5.3 Selecting the Explanation

We have already described how to derive and train a neural network from a valid explanation based on qualitative domain theory. However, given a set of qualitative assertions, many valid explanations (based on different subsets of the knowledge) may be possible. Simply using all qualitative assertions in the background knowledge may lead to an overly complex neural network and poor generalization to test examples. In this section we describe an iterative greedy procedure to select a single valid explanation, corresponding to a subset of the available qualitative background knowledge.

Assuming that our complete set of qualitative assertions is valid, we can construct a valid *maximal explanation* based on directed acyclic graph corresponding to all qualitative assertions. This assumption holds in our case, leading to the maximal explanation shown in Figure 3. The approach proceeds by iteratively training a neural net, pruning off the "weakest" assertions in the explanation structure, and repeating until all qualitative assertions have been eliminated. Finally, the explanation structure that performed best on a set of withheld validation data is output.

1. $X \leftarrow$*maximal explanation*
2. While X contains one or more edges:
   (a) Construct and initialize neural network $\nu(X)$
   (b) Train $\nu(X)$ using back-propagation with training set M
   (c) $error(\nu(X)) \leftarrow$ the error of $\nu(X)$ on validation data
   (d) For each edge $f \rightarrow f' \in \nu(X)$: $effect(f \rightarrow f') \leftarrow (\frac{1}{|M|}\sum_{m \in M} \frac{\partial safety}{\partial f'})w(f,f')\sigma(f)$
   (e) Remove edge $\arg\min_{f \rightarrow f' \in \nu(X)} effect(f \rightarrow f')$ and subsequent edges and features required to make X valid.
   (f) For any features f with one incoming edge $f' \rightarrow f$ remove f and replace all edges $f \rightarrow f''$ with $f' \rightarrow f''$
3. Return $\nu(X)^* = \arg\min_{\nu(X)} error(N)$

where $\sigma(f)$ represents the standard deviation of f across all training examples. At step (d), we identify the edge in the neural network that has the least effect on the output value. We eliminate this edge (equivalent to eliminative the corresponding qualitative assertion from our working set of background knowledge), prune additional structure if the resulting explanation is invalid (if non-native features are left undefined, or if nodes are left unattached to the output value), collapse redundant features, and repeat. Details of the cross validation procedure appear in the evaluation section.

6 Evaluation

Dataset. We utilized a set of 100 mission scenario based on military air operations context referenced at the beginning of the paper. All scenarios are set at the Iran/Iraq border, with 6 fixed unfriendly missile locations (Red). A random automated procedure is used to place the relevant airspaces (Blue: Combat Air Patrol (Protector), Green: High Value Asset (Protectee), Orange: Unfriendly Threat) within the region of interest. An example placement was shown in Figure 2. These placements are subject to plausibility constraints designed by our domain expert. Additional features such as threat level, priority level, aircraft types, and number of aircraft are chosen randomly from the available options.

These scenarios were presented for scoring to our domain expert. Scenarios were scored on an integer scale, 1 (very unsafe) to 5 (very safe). To be compatible with our neural network implementation, these scores were rescaled between 0 and 1. The 31 native features described in our domain theory are computed and normalized to have mean zero and variance one across all examples, before being presented to our learning algorithms.

Table 1: Average squared error for our approach (Explanation-Based NN) and the Hidden-Layer NN over test examples.

Training Examples	20	40	60
Explanation-Based NN	.0534	.0386	.0321
Hidden-Layer NN	.0601	.0481	.0409

Setup. We compare our approach to a standard hidden layer neural network (HLNN) technique [Bishop, 2005]. In HLNNs, a neural network is constructed consisting of three layers: an input layer with one node per native feature, a hidden layer with some number of nodes (2 to 15 in our experiments), and an output layer with one node corresponding to safety value. Each node in the input layer has an edge to each node in the hidden layer, and each edge in the hidden layer has an edge to output layer. These hidden layer neural nets are trained using the same back propagation procedure as is used in our approach.

For training, we use cross validation to resolve the neural network/explanation structure. In our approach, the full explanation structure is iteratively pruned yielding a number of competing neural network structures from which one must be chosen. In the HLNN approach, the number of hidden units must be chosen. Cross validation is also used to avoid overfitting via the technique of early stopping [Sarle, 1995]. In this approach, during training, the neural network is continuously evaluated against a validation set of data, and the neural network weights that yielded the best score on the validation set are applied to the test data.

We accomplish both of these goals at once as follows. The training data is split into 5 separate training/validation sets. Each candidate network structure is initialized (according to the qualitative background for our approach, with random weights from [-1,1] for the hidden layer approach), trained separately on each split, and evaluated at each training iteration against the validation data. Across all structures/training iterations, the network with the best average performance across all five validation data sets is selected for application to the test data. For the chosen network structure, each test example is labeled with the average network output across each of the five trained network weight sets.

Results. We test the two approaches for 20, 40 and 60 training examples. The remaining examples are used for evaluation. Data is randomly split between training/testing sets and 10 trials are performed for each size. The average squared error of each approach appears in Table 1. Across all training set sizes, the explanation-based neural network approach outperforms the hidden-layer approach by 16.3%. For each of 40 and 60 training examples, a paired t-test suggests that the explanation-based approach outperforms the hidden-layer approach with probability greater than .975. The performance of the explanation-based network at 40 examples exceeds that

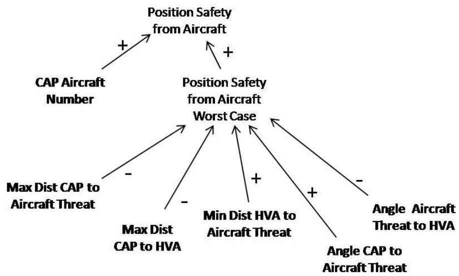

Figure 4: The most commonly selected sub-explanation for "Position Safety from Aircraft."

of the hidden-layer neural network at 60 examples, suggesting that the "value" of the qualitative domain theory is greater than that of 20 additional training examples, a substantial (50%) increase in the size of the training set.

Interesting is the most commonly selected explanation structure. In this explanation, all qualitative assertions remain as depicted in Figure 3 except for those beneath the definition of "Position Safety from Aircraft". The original qualitative knowledge accounts for the fact that when only 2 aircraft are assigned to the CAP, they fly together, and thus must be in position to defend against a threat no matter where they are located in the airspace. However when 4 aircraft are present, they split up into two sets, and thus some aircraft is usually well positioned within the airspace to defend against a threat. The learning procedure prunes away much of this knowledge, settling on the sub-explanation that is appropriatly complex given the training examples. This sub-explanation is depicted in Figure 4. In this definition, only the worst case distances are used, although aircraft number still positively effects "Position Safety from Aircraft," as it should. This is still a coherent knowledge structure, and the reduced complexity of the resulting neural network structure likely leads to better generalization.

7 Related Work

Previous work on developing prediction/learning systems for airspace operations includes the Causal Analysis Tool (CAT) from the Air-Force Research Laboratories (AFRL) for use in creating, modifying and analyzing causal models of airspace operations. CAT's basic function is to propagate local estimates of uncertainty throughout large models, estimating the probability, as a function of time, that particular events will be true. For details see [Lemmer, 1996].

Another system for airspace operations that has been under development recently is the Generalized Integrated Learning Architecture (GILA) [Oblinger, 2005]. This system is an ensemble planning and learning system developed as part of a large team effort, and funded by DARPA. The emphasis in GILA is to develop an AI system that consists of loosely- coupled learner/planner components. The present system grew from GILA's learning and model-checking system for safety constraints in airspace operations. See [Rebguns *et al.*, 2008] for details.

Our learning approach described in this paper has its roots in the KBANN formalism of [Towell and Shavlik, 1994]. Like KBANN, our learning procedure is based on a technique to translate a set of domain-theory rules into ANNs and train those ANNs in order to learn a prediction on an objective feature of the targeted domain. One important difference between KBANN and our approach here is the kind of the domain theory translated into ANNs: KBANN uses propositional Horn clauses to describe relationships between the facts of a problem, whereas we use representations from qualitative process theory. Additionally, KBANN hypothesizes a fully connected ANN while our structure is more sparese, constrained by the explanation.

Another relevant work is on Explanation-Based Neural Networks (EBNN) described in [Thrun and Mitchell, 1993]. In the EBNN framework the domain theory is described in terms of neural networks, rather than qualitative inferential rules. Furthermore, the EBNN structure is directly employed rather than serving as a safety filter for other systems. We also found it convenient to generate the maximal explanation of a goal concept and then apply a pruning procedure to find the greedy best subset that simultaneously best respects both the training examples and the domain theory.

Finally, our approach meshes well with the "Deep Learning Architectures" promoted in [Bengio and LeCun, 2007]. In Deep Learning, Lecun argues that function-approximation learning methods can be substantially improved by the judicious use of many hidden layers, provided they possess certain invariance properties. Our work can be seen as a type of "deep learning" approach since we are generating multilayered ANN models, and the domain theory achieves the desired conceptual invariance.

8 Conclusions

In this paper, we have described our automated learning/reasoning AI system for producing greedy-best explanations for a target concept in a problem domain, given a domain theory and only a few training examples. The system assumes the domain theory is represented in a sub-language of the well-known QP Theory. Given this form of domain theory and some examples in that domain, the system first generates the maximal explanation of the target concept and then exploits a pruning approach to extract a greedy-best explanation out of the space of explanations.

Although our approach is general, i.e., it can be applied in any problem domain, we focussed in this paper on airspace operations and mission safety (our target concept) in those operations. Our experimental evaluation demonstrates the effectiveness of our approach to standard hidden-layered neural network based learning techniques.

As a future work, we intend to perform an extensive theoretical and experimental evaluation of the system. We also plan to investigate bottom-up techniques for building explanation structures. This would make our approach amenable to domains where huge amounts of background knowledge are available.

Acknowledgments. This work is funded by the DARPA GILA Contract # FA8650-06-C-7605. The opinions expressed in this paper are those of authors and do not necessarily reflect the opinions of DARPA.

References

[Bengio and LeCun, 2007] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards ai. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large-Scale Kernel Machines*. MIT Press, 2007.

[Bishop, 2005] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 2005.

[Forbus, 1997] Kenneth Forbus. Qualitative reasoning. In *The Computer Science and Engineering Handbook*, pages 715–733. 1997.

[Kuipers, 1994] Benjamin Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, 1994.

[Lemmer, 1996] John Lemmer. The causal markov condition, fact or artifact. *SIGART Bulletin*, 7(3):3–16, 1996.

[Oblinger, 2005] D. Oblinger. Darpa transfer learning program: Proposer information pamphlet, 2005. http://www.darpa.mil/ipto/solicitations/closed/05-29_PIP.htm.

[Rebguns *et al.*, 2008] A. Rebguns, D. Green, G. Levine, U. Kuter, and D. Spears. Inferring and applying safety constraints to guide an ensemble of planners for airspace deconfliction. In *COPLAS 2008: CP/ICAPS 2008 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems*, 2008.

[Sarle, 1995] W. S. Sarle. Stopped training and other remedies for overfitting. In *27th Symposium on the Interface of Computing Science and Statistics*, pages 352–360, 1995.

[Thrun and Mitchell, 1993] Sebastian Thrun and Tom Mitchell. Integrating inductive neural network learning and explanation-based learning. In R. Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1993.

[Towell and Shavlik, 1994] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70:119–165, 1994.

[Wikipedia, 2009] Wikipedia. Risk (game). http://en.wikipedia.org/wiki/Risk_(game), May 2009.