Weights of Evidence for Intelligible Smart Environments

Brian Y. Lim, Anind K. Dey
Human-Computer Interaction Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
{byl, anind}@cs.cmu.edu

ABSTRACT

Smart environments are improving their performance and services by increasingly using ubiquitous sensing and complex inference mechanisms. However, this comes at a cost of reduced intelligibility, user trust and control. The Intelligibility Toolkit was developed to support the automatic generation and provision of explanations to help users understand context-aware inference. We have extended the toolkit to generate explanations for a wider range of inference models and to provide two styles of explanations — rule traces and weights of evidence. We describe explanations generated from several inference models for a smart home dataset for activity recognition. This demonstrates the versatility of using the Intelligibility Toolkit to retain explanatory capabilities across different inference models.

Author Keywords
Context-awareness, intelligibility, explanations, toolkits.

ACM Classification Keywords
H.m. Information systems: Miscellaneous.

General Terms
Algorithms, Human Factors.

INTRODUCTION

Smart environments increasingly use diverse sensors and various mechanisms for reasoning and inference to provide more appropriate services to end-users. This complexity can be difficult for end-users to understand, leading to a loss of trust and control in these systems [2]. Therefore, smart environments need to be intelligible [6] by automatically providing explanations of application behavior. To support the development of intelligible smart environments, several software toolkits and frameworks have been developed (*e.g.*, [1, 3, 7, 15]). This paper adds to this body of work by extending the Intelligibility Toolkit [7] to support more Explainers for a wider range of inference models. In particular, we standardize explanations into two styles (rule traces and weights of evidence) to simplify the generation of explanations for Why and Why Not questions.

We validate the Explainers by demonstrating several generated explanations from a smart home dataset. We show that the weights of evidence explanation style has the potential to provide a consistent explanation interface for explaining how smart environments make decisions, independent of the underlying inference model used. However, discrepancies between the explanations across inference models can subtly affect user understanding.

UbiComp'12, September 5-8, 2012, Pittsburgh, USA.
ACM 978-1-4503-1224-0/12/09.

INTELLIGIBILITY TOOLKIT

The Intelligibility Toolkit [7] has several components to support the automatic generation and provision of explanations from context-aware inference: **Explainers** which contain algorithms to generate explanations based on the application inference model, **Queries** to encapsulate questions that end-users may ask of the smart environment, **Explanation Expressions** to contain information of the generated explanations from Explainers, **Reducers** to simplify the explanation Expressions using various heuristics or mechanisms, and **Presenters** to render the final explanation data structure in a human consumable format (*e.g.*, text or visualizations). The Intelligibility Toolkit is extensible to support new explanation types, inference models, reduction heuristics, presentation styles and formats, and explanation selection criteria. The toolkit has been used to build several demonstration applications [7], and more complex intelligible context-aware applications (*e.g.*, [8, 10]). We expand support for model-based explanations and standardize two explanation styles.

Model-Based Explanation Question Types

The Intelligibility Toolkit provides explanations to various question types end-users may ask as described [6]. In this paper, we focus on two *model-based* explanations that explain the mechanism or reasoning process used in the context-aware application. These depend on the inference model used and they answer the questions:

1. **Why** is this context inferred as the value *X*?
2. **Why Not**: why isn't this context inferred as *Y*, instead?

Smart environments and context-aware applications use many different types of inference models to be more intelligently aware of the user and the environment. We have expanded the support for explaining inference models from four to 10, including: rules, decision trees, functions (linear regression, logistic regression, support vector machines), Bayesian models (naïve Bayes, hidden Markov models), similarity models (k-nearest neighbors), and ensemble methods (Bagging, AdaBoost). We define two styles in which explanations may answer Why and Why Not questions, which we describe next.

Styles of Explanations

Differences in inference models affect how explanations are generated. The Intelligibility Toolkit currently supports two explanation styles: rule traces and weights of evidence.

Rule traces describe the line of reasoning to explain Why an inference was made. A trace is represented as a conjunction of literals (*e.g.*, Hour > 6 AND Occupancy > 0). Why Not explanations are provided as traces for alternative inferences that were not executed. The toolkit provides these explanations for Rules and Decision Trees.

Many models do not make inferences using rules (*e.g.*, naïve Bayes, SVM), so rule traces are not relevant for explaining them. Instead, we employ the **weights of evidence** style of explanation also used in [5, 12, 13]. This considers that the model computes a *total* evidence for each possible outcome value that may be inferred, and that this total evidence is due to a sum of *atomic* weights of evidence due to various Input factors. Therefore, this explains to the user how much evidence each factor contributes towards or against the inference. We next describe how Why and Why Not explanations may be generated from some inference models.

ALGORITHMS FOR WEIGHTS OF EVIDENCE

We describe the basis of the weights of evidence explanation style as an *absolute evidence* due to a sum of weights, and how Why and Why Not explanations can be derived from this. These weights may be due to the input feature values voting for or against an inference. Depending on the inference model, there may also be more *dimensions* of atomic weights of evidence. For example, ensemble classifiers such as Bagging or Boosting have classifier iterations as a dimension in addition to input features.

Absolute Evidence

We represent the evidence for inferring the ith class as:

$$g_i = \sum_{r \in R} f_{ir} \tag{1}$$

where f_{ir} is the rth atomic weight of evidence and R is the space of all atomic weights. Equation (1) requires that the explainer is able to derive a *linear additive* expression of atomic units. This is easy for linear classifiers (*e.g.*, linear SVM), but in general, isotonic (monotonic increasing) transformations may be required (*e.g.*, naïve Bayes). With these *absolute* weights of evidence, we can derive weights of evidence explanations for Why and Why Not questions.

Why Not Explanation

This explains why the jth class not inferred, but the ith was:

$$\Delta g_{ij} = g_i - g_j = \sum_r \Delta f_{ijr} \geq 0 \tag{2}$$

where $\Delta f_{ijr} = f_{ir} - f_{jr}$ and we assume that the atomic weights of evidence are separable by each atomic unit. Note that the jth class may have been inferred, but just not with the highest certainty among all class values.

Why Explanation

This explains why the ith class was inferred over all m other class values. Consequently, Equation (2) holds for $\forall j$, such that we can sum over $\forall j$ and normalize over class values to get the Why explanation:

$$\Delta g_{i\forall} = \frac{1}{m} \sum_{j=1}^{m} \Delta g_{ij} = \sum_{j=1}^{m} \sum_r \frac{\Delta f_{ijr}}{m} \geq 0 \tag{3}$$

Equations (2), and (3) are implemented in the base Weights of Evidence Explainer. Rather than developing Why and Why Not explanation algorithms for Explainers of new inference models, developers only need to derive an expression for Equation (1) or (2). Next, we briefly describe derivations of Equation (1) for 3 inference models.

Naïve Bayes Explainer

For naïve Bayes, the posterior probability of inferring the ith class $(y = y_i)$ from a set of m class values, given the observed instance input feature values x, is a product of prior probability and feature likelihoods:

$$P(y_i|x) = P(y_i) \prod_{r=1}^{n} P(x_r|y_i) \tag{4}$$

where x_r is the rth input feature value and n is the number of features. We can derive a linear additive expression suitable for Equation (1) by taking a log of Equation (4):

$$g_i = \log(p_i) = \sum_{r=0}^{n} \log(p_{ir}) = \sum_{r=0}^{n} f_{ir} \tag{5}$$

where $f_r = \log(p_{ir})$. Hence with Equation (5), naïve Bayes can be explained as the sum of evidence:

1. Prior probabilities of selected class value $(r = 0)$
2. Due to each feature value $(r > 0)$

Decision Tree Explainer

We can derive weights of evidence explanations for decision trees that are trained using statistical techniques over a training set (*e.g.*, J48, Random Tree). We first consider the evidence in a trace, rather than the full input feature set. A reasoning trace of length η is represented by the conjunction:

$$Y_i \cap X_1 \cap X_2 \cap \ldots \cap X_\eta = Y_i \cap \bigcap_{\rho=1}^{\eta} X_\rho$$

where Y_i is the event that the ith class is inferred and X_ρ is the ρth condition literal in the trace. A condition may be an equality or inequality, *e.g.*, $x_r = 10$, $x_r > 7$, where x_r refers to the rth input feature.

If we approximate that each condition is independent of one another given the rule trace, then the probability of inferring the ith class can be expressed as:

$$P\left(Y_i \cap \bigcap_{\rho=1}^{\eta} X_\rho\right) \cong P(Y_i) \prod_{\rho=1}^{\eta} P(X_\rho) \tag{6}$$

For notational convenience, we rewrite Equation (6) as

$$p_i \cong \prod_{\rho=0}^{\eta} p_{i\rho} \tag{7}$$

where $p_{i0} = P(Y_i)$ is the prior probability for inferring the ith class and $p_{i\rho} = P(X_\rho)$ for $\rho > 0$. p_i is just the probability certainty of the inference. $P(Y_i)$ is estimated from the probability class distribution of the full training dataset. The class probability distribution at the ρth node in the decision tree equals the product of probabilities due to conditions from the root down to the ρth node, *i.e.*, $P\left(\bigcap_{q=0}^{\rho} X_q\right) \cong \prod_{q=0}^{\rho} P\left(X_q\right)$. So, we can compute $P\left(X_\rho\right)$ using the probability distribution from the $(\rho-1)$th parent node:

$$P\left(X_\rho\right) = \frac{P\left(\bigcap_{q=0}^{\rho} X_q\right)}{P\left(\bigcap_{q=0}^{\rho-1} X_q\right)} \tag{8}$$

Taking a log transform of Equation (7) gives the weights of evidence explanations of a trace inferring the ith class:

$$g_i = \sum_{\rho=1}^{\eta} f_{i\rho} \tag{9}$$

where $f_{i\rho} \cong \log\left(p_{i\rho}\right)$ and η is the trace length.

To get the weights of evidence in terms of each input feature, x_r, we sum evidences of the same feature together:

$$g_i = \sum_{r=0}^{n} f_{ir} \tag{10}$$

where $f_{ir} = \begin{cases} \log(p_{i0}) & \text{, if } r = 0 \\ \sum_{\rho=1}^{\eta} \log\left(p_{i\rho}\right) [\![x_r \in X_\eta]\!] & \text{, if } r > 0 \end{cases}$

$$[\![x_r \in X_\eta]\!] = \begin{cases} 1 & \text{, if } X_\eta \text{ describes } x_r \\ 0 & \text{, otherwise} \end{cases}$$

This is useful for explainers of ensemble classifiers.

Bagging Explainer

Bootstrap aggregation (bagging) is an ensemble classification algorithm which trains C versions of a base classifier (*e.g.*, J48), one classifier for each bootstrapped training set. The ensemble inference is performed by averaging the inference of each of the base classifiers:

$$p_i = \frac{1}{Z} \sum_{c=1}^{C} p_{ic} \tag{11}$$

where p_{ic} is the probability of the cth classifier inferring the ith class, and Z is a normalization constant. We can express the weights of evidence for inferring the ith class as:

$$g_i = \sum_{c=1}^{C} p_{ic} \frac{g_{ic}}{\text{value}(g_{ic})} \tag{12}$$

where g_{ic} is the linearly separable weights of evidence expression for the cth base classifier and $\text{value}(g_{ic}) = \text{sgn}(g_{ic})|g_{ic}|$ is the value of the total weights of evidence of g_{ic} used to normalize the weights for each base evidence.

Similarly, we can derive the weights of evidence for the Random Forest classifier which is a bagged ensemble of Random Trees, and for the AdaBoost meta classifier.

DEMONSTRATION APPLICATION

We demonstrate several explanations that can be generated with the Intelligibility Toolkit using different inference models for a smart home activity recognition application. We formatted the dataset of [14] to the Weka ARFF format and trained several classifiers (inference models) using the Weka toolkit [4]. Our purpose is to explore the similarity and variance in explanations, not necessarily to train classifiers of high accuracy. For clarity, we used non-temporal classifiers, although the Intelligibility Toolkit can generate explanations for hidden Markov models (see [7]). We also limited the features to a relatively small set to be illustrative. For this application, classifiers were trained to infer one of 7 domestic activities. The 28-day dataset was split into a training (first 14 days) and a test (remainder) set. Figure 1 shows how a weights of evidence explanation may be visualized as a floorplan to explain the inferred domestic activity. Table 1 shows how different inference models lead to similar weights of evidence explanations for the same inference, but with some discrepancy. Therefore, this leads to interesting research questions for investigation which we leave for future work.

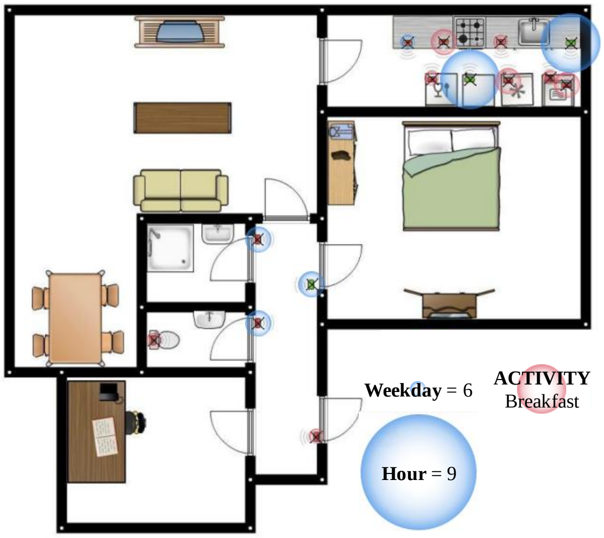

Figure 1: Why explanation visualization from an application using a naïve Bayes classifier to model domestic activity. This explains why the user Activity was inferred as Breakfast. Evidence due to features are indicated by the area of bubbles around the corresponding sensors in the floorplan, non-physical inputs (Weekday and Hour), and evidence due to the prior probability (Activity). Blue bubbles indicate evidence voting for the inference and red bubbles indicate evidence voting against. We can see that the Hour = 9am is a strong indicator of inferring the Activity as Breakfast, while the red bubble for Activity indicates this is inference is unlikely, given no other input information.

CONCLUSION AND FUTURE WORK

Providing explanations can help context-aware applications and smart environments be more intelligible to help increase user understanding and trust. We have presented improvements to the Intelligibility Toolkit to support a wider range of popular inference models for intelligent and context-aware systems. The Intelligibility Toolkit aims to make it easier for developers to provide many explanation types in their context-aware and smart environments. In particular, we have generalized the support for the *weights*

		Rule Trace	Weights of Evidence					
Feature	**Value**	**DT**	**DT**	**NB**	**SVM**	**Bg.DT**	**RF**	**AB.DT**
ACTIVITY	Breakfast		-127	-125	-4779	-1	-1	-96
Front Door	0		0	-2	0	0	70	0
Hall-Bedroom Door	1		0	16	-5050	0	123	-81
Hall-Toilet Door	0		0	14	0	120	228	350
Hall-Bathroom Door	0		0	10	0	0	230	0
Toilet Flush	0		0	-5	0	0	57	0
Fridge	1	= 1	240059	182	4511	720	1405	356
Freezer	0		0	-6	0	0	0	0
Microwave	0		0	-20	0	0	488	0
Cups cupboard	0		0	8	0	0	0	0
Plates Cupboard	0		0	-33	0	0	432	0
Pans Cupboard	0		0	1	0	0	67	0
Grocery Cupboard	1		0	211	9463	520	783	1000
Dishwasher	0		0	-2	0	0	0	0
Washing Machine	0		0	-2	0	0	0	0
Weekday	6		0	10	130	0	258	317
Hour	9	8 < Hr ≤ 13	560000	1342	-3808	2560	179	1648

DT=decision tree, NB=naïve Bayes, SVM=linear support vector machine, Bg.DT=bagged DT, RF=Random Forest, AB.DT=AdaBoosted DT

Table 1. Why explanations in Rule trace and weights of evidence styles of explanations for different classifiers for a specific test instance where Activity was inferred as Breakfast for all classifiers. The Hall-Bedroom Door, Fridge, and Grocery Cupboard were open, Weekday was Saturday (=6), and Hour was 9 (*i.e.*, between 9 and 10am). The rule trace of the decision tree had 3 conditions of 2 factors. Note that the weights of evidence are in terms of Inputs ($r > 0$) and Output ($r = 0$) values. The weights of evidence across classifiers are not to scale. Nevertheless, they can easily be substituted into Figure 1 to explain their respective classifiers.

of evidence explanation style to explain model-based questions, Why and Why Not, in terms of weights due to input factors. This ease can allow developers to perform rapid prototyping of different explanation types to discern the best explanations to use and the best ways to use them. By standardizing the styles of explanations, developers have many more choices when selecting classifiers to increase application accuracy and performance, while retaining the intelligibility features of their application and keeping unchanged the explanation interfaces.

Weights of evidence explanations can help end-users to identify how intrinsic input factors and sensors influence the decision and inference in smart environments. However, the differences between inference models can lead to slight differences in the generated explanations, and this can variously influence the user's understanding. Therefore, for future work, we intend to compare and evaluate the intelligibility of different inference models on user understanding. We intend to objectively quantify the discrepancy between explanations of different inference models, and determine whether the discrepancy will decrease as the performance and accuracy of each inference model increases, and the models converge to a similar true concept. Finally, we intend to conduct a user study to investigate how successfully lay end-users can interpret the weights of evidence explanations and how differences in explanations affect their understanding of the same inference.

REFERENCES

1. Assad, M. *et al.* (2007). PersonisAD: Distributed, Active, Scrutable Model Framework for Context-Aware Services. *Pervasive 07*, 55-72.
2. Barkhuus, L. & Dey, A.K. (2003). Is context-aware computing taking control away from the user? Three levels of interactivity examined. *Ubicomp 03*, 149–156.
3. Dey, A.K. & Newberger, A. (2009). Support for context-aware intelligibility and control. *CHI 09*, 859-868.
4. Hall, M. *et al.* (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations 09*, 11(1), 10-18.
5. Kuleza, T. *et al.* (2009). Fixing the Program My Computer Learned: Barriers for End-users, Challenges for the Machine. *IUI 09*, 187-196.
6. Lim, B.Y. & Dey, A.K. (2009). Assessing Demand for Intelligibility in Context-Aware Applications. *Ubicomp 09*, 195-204.
7. Lim, B.Y., Dey, A.K. (2010). Toolkit to Support Intelligibility in Context-Aware Applications. *Ubicomp 10*, 13-22.
8. Lim, B.Y. & Dey, A.K. (2011). Design of an Intelligible Mobile Context-Aware Application. *MobileHCI 11*, 157-166.
9. Lim, B.Y. & Dey, A.K. (2011). Investigating Intelligibility for Uncertain Context-Aware Applications. *Ubicomp 11*, 415-424.
10. Lim, B.Y. & Dey, A.K. (2012). Evaluating Intelligibility Usage and Usefulness in a Context-Aware Application. *CMU-HCII Technical Report*.
11. Lim, B.Y. (2012). Improving Understanding and Trust with Intelligibility in Context-Aware Applications. Ph.D. Thesis, May 2012, Carnegie Mellon University.
12. Mozina M. *et al.* (2004). Nomograms for Visualization of Naive Bayesian Classifier. *PKDD 2004*, 337-348.
13. Poulin, B. *et al.* (2006). Visual explanation of evidence in additive classifiers. *IAAI 06*, 1822-1829.
14. van Kasteren, T.L.M. *et al.* (2008). Accurate Activity Recognition in a Home Setting. *Ubicomp 08*, 1-9.
15. Vermeulen, J. *et al.* (2010). PervasiveCrystal: Asking and Answering Why and Why Not Questions about Pervasive Computing Applications. *IE 10*, 271-276.