Justification Narratives for Individual Classifications

Or Biran ORB@CS.COLUMBIA.EDU
Kathleen McKeown KATHY@CS.COLUMBIA.EDU
Department of Computer Science, Columbia University, New York, NY 10027

Abstract

Machine learning models are now used extensively for decision making in diverse applications, but for non-experts they are essentially black boxes. While there has been some work on the explanation of classifications, these are targeted at the expert user. For the non-expert, a better model is one of justification - not detailing how the model made its decision, but justifying it to the human user on his or her terms. In this paper we introduce the idea of a justification narrative: a simple model-agnostic mapping of the essential values underlying a classification to a semantic space. We present a package that automatically produces these narratives and realizes them visually or textually.

Keywords: Explanation, Justification

1. Introduction

The need for explaining and justifying automatically-generated predictions has been discussed in various contexts, beginning with expert systems as early as the 1970's (Shortliffe and Buchanan, 1975; Swartout, 1983). It is particularly important in high-risk applications such as medicine: in an early study, physicians rated the ability to explain decisions as the most highly desirable feature of a decision-assisting system (Teach and Shortliffe, 1981). It is also essential in consumer-facing applications such as Recommender Systems (Herlocker et al., 2000; Tintarev and Masthoff, 2007) and Context-Aware Applications (Lim and Dey, 2010; Vermeulen, 2010).

The terms *explanation* and *justification* are sometimes used interchangeably, but the distinction is important when considered from the point of view of a non-expert. Explanation answers the question "how did the system arrive at the prediction?" while justification aims to address a different question, "why should we believe the prediction is correct?". For a user with enough expertise, an answer to the first question will suffice to satisfy the second. Consequently, explanation of classification results in the machine learning literature has often been treated as the visualization problem of providing detail about the internals of the model in a human-consumable form (Szafron et al., 2003) or as the problem of projecting the processes and parameters of non-linear models to make them more meaningful to the expert (Robnik-Sikonja and Kononenko, 2008; Baehrens et al., 2010), while justification as a goal has not been mentioned.

In machine learning, unlike rule-based or knowledge-based expert systems, it is not reasonable to expect non-experts to understand the details of how a prediction was made. It is still important, however, that they understand the variables affecting the current prediction enough to satisfy the question of justification. It has been shown that evidence-based causal models of justification are often more satisfactory to users than full white-

box models, and that replacing numeric values with qualifying linguistic expressions (high, strong, etc) also enhances satisfaction (Lacave and Díez, 2002; Herlocker et al., 2000). The independent variables used in machine learning models often correspond to real-world evidence that non-experts understand well, and a justification for a prediction can rely on these variables, their importance to the model, their effect on the prediction, and their interactions.

A robust method of automatically generating prediction justification for non-experts, then, should focus on selecting the most important pieces of evidence for a particular prediction and on analyzing and presenting their roles in the prediction. The selected evidence should be presented to the user in a way that is invariant across different models and readily understandable by a non-expert.

In this paper we will discuss some central concepts of explanation and how they can be used without model-specific detail in justification. We also introduce the idea of a justification narrative and propose a framework of deriving it from a single prediction. Finally, we present PreJu (the Prediction Justifier): a package that produces these structured narratives and uses them to generate visual or textual justifications for predictions.

2. Core Concepts of Explanation

While proposed methods of explanation in the literature vary significantly across models and authors, two concepts in particular appear in many of them under different names. These are the *effect* of a feature on the prediction and the feature's *importance* in the model.

We use Logistic Regression (LR) as an example throughout this paper. The effect of each individual feature on the prediction in LR is clear and independent of other features, and it is easy to define the importance of features, all of which is ideal for explaining the justification narrative. While for many other models it is harder to define the values of these scores (in particular, the values in non-linear models may differ depending on the values of other features), there have been specific proposals for various models (Carrizosa et al., 2006; Yap et al., 2008) as well as proposals for model-agnostic definitions (Robnik-Sikonja and Kononenko, 2008). We view this work as complementary and essential to ours. This paper is not focused on dealing with particular models, but on creating a justification narrative structure, which can be done through abstracting the details behind these two scores. We can be agnostic to the origin of the values as long as they have normal mathematical equality and inequality relations (greater than, less than, and equals). The more the values correspond to the true workings of the model, the better the justification will be.

A typical formulation of Logistic Regression predicts a class $y \in (y_1, y_2, ...y_k)$ for an instance x by applying the logistic function:

$$Pr(y = y_j|x) = \frac{1}{1 + e^{-\sum_i \theta_{ji} x_i}}$$

Where each x_i is a feature value and the coefficients θ_i have been learned by linear regression from the training data for each class j, and include an intercept. Since this is a linear model, we can define the fixed *polarity* of a feature i towards a class j as the sign function of its coefficient, $sgn(\theta_{ji})$.

The **effect** (sometimes called *contribution*) of a feature is the extent to which it has contributed towards or against predicting a particular class for the instance. Since we

are interested in justifying the current prediction, we will define the effect as being the contribution with regard to the predicted class only.

For a linear model, the effect of a feature i towards class j is the product of the feature's coefficient and its value in the instance:

$$\text{Effect}_{ji} = \theta_{ji} x_i$$

We use **importance** to model the overall strength of a feature in the model. One way to think of it is as the *expected* effect of the feature on the prediction for a particular class, which in a linear model can be estimated using the mean feature value for the class (X^j is the set of all instances in the data set with class j):

$$\text{Importance}_{ji} = \theta_{ji} \frac{\sum_{x \in X^j} x_i}{|X^j|}$$

Like polarity, importance in LR is fixed across different instance predictions. Note that the importance does not necessarily have to have the same sign as the polarity; if negative feature values are allowed, the signs may be different.

3. Justification

Building on the two concepts discussed in the previous section, we introduce the idea of a *justification narrative*.

A justification narrative is defined for each instance prediction, and is composed of a subset of the model features called *key features*, each of which has a narrative *role*. The task of building a justification narrative is the task of selecting the key features and assigning their roles.

3.1. Narrative Roles

Narrative roles are assigned based on the sign and magnitude of the importance and effect of a feature towards the predicted class. They represent semantically clear concepts that non-experts readily understand, and are rooted in the true details of the prediction. Table 1 shows the roles for all possible combinations.

The magnitude of the importance of the features is a simple separation between *low* and *high*. The separation is dependent on the classifier, and can be done in various ways. A few simple options we have implemented are a fixed threshold; a fixed number of features (with the highest importance) that are considered high, while all others are considered low; and a scheme where the next feature with the highest importance is iteratively considered high until the sum of importance of all high importance features is more than some percentage of the total sum of importance. These schemes are all done separately for positive and negative importance features. The magnitude of effect is determined with a threshold that is set midway between the highest low importance and the lowest high importance.

The method presented in this paper was developed in the context of a large system we built to predict the prominence of scientific topics. We will use a simplified version of that task and prediction system as an example throughout the rest of this section to illustrate the various roles.

We have a (trained) binary Y/N classifier that predicts whether or not the citation count of a scientific article will increase in the next year. Each instance is an article, and we

Importance \ Effect	High positive	Low	High negative
High positive	Normal evidence	Missing evidence	Contrarian counter-evidence
Low	Exceptional evidence	Negligible	Exceptional counter-evidence
High negative	Contrarian evidence	Missing counter-evidence	Normal counter-evidence

Table 1: Narrative roles assignment for the range of feature effect and importance

extract (among others) four features. Since the model is trained, we know the importance of each of the features towards the Y class:

SLOPE is the slope of the linear function of the article's citations per year over the past 5 years (all articles in the data set are at least 5 years old). It models the recent increase or decrease in the number of citations and has a high positive importance.
H-INDEX is the maximum h-index of the authors. It has a low positive importance (people have a slight preference towards citing prominent authors).
VENUE is a score between -1 and 1 that represents the prominence of the venue the article was published in. Although its polarity is positive, it has a low negative importance (because the average article is published in a less-than-average, negative-score venue).
DIAMETER is the diameter of the *citation network*, built by treating all articles which previously cited the instance article as nodes and the citations between them as directed edges. It has a high negative importance (large diameters suggest that the article is cited not by a community, but by unconnected authors)

Now, consider a particular article which the model classifies as Y (its citation count is predicted to increase in the next year). We will use possible effects of the features to explain each of the roles below.

First, we describe evidence roles. Features having these roles contributed towards the prediction. **Normal evidence** is evidence that is expected to be present in many instances predicted to be in this class: with high importance, high effect is not surprising. For our Y prediction, SLOPE having a high positive effect would be normal evidence. **Exceptional evidence** is evidence that is not usually expected to be present. With low importance, high effect means that the feature value is exceptionally high. In our example, if H-INDEX has a high positive effect, it means at least one of the authors is exceptionally prominent. **Contrarian evidence** is strongly unexpected, since the effect has the opposite sign than expected. In our example this is impossible, because DIAMETER cannot be negative.

Features that have **Missing evidence** as a role are important features that were expected to contribute positively, but were weak for the particular instance. If SLOPE had a low effect in the prediction, that is something we want to include in the narrative because it means that the prediction was uncharacteristically made without the important effect of that feature. Similarly, **missing counter-evidence** is given to features that were expected to contribute negatively but did not.

Finally, there are counter-evidence roles. Features having these roles contributed against the prediction, although they were not strong enough to prevent it. **Normal counter-evidence** is expected: it is normal for DIAMETER to have a high negative effect (which means the article has an average DIAMETER value) even if the positive effects from (say) SLOPE or H-INDEX ultimately overpower it. **Exceptional counter-evidence** is not expected - if VENUE has a high negative effect, then the venue is exceptionally poor. Finally, if SLOPE has a high negative effect (the article is one of the rare ones for which the number of citations per year decreases over time), we have **contrarian counter-evidence** because the feature we normally expect to contribute positively contributes negatively instead.

Note that contrarian evidence (and contrarian counter-evidence) is only possible for features that may have negative values, and may not appear in many real-world applications.

3.2. Key Feature Selection

The appropriate way to select the key features depends on the application. If the feature space is small and/or sparse, it may make sense to select all features that have any role other than negligible. Some consumer-facing applications (e.g., recommender systems) may want to select a fixed number of features to show as evidence to the user, and may want to constrain their roles (only showing the *evidence* roles in the first column of table 1, for example). In other cases, we may want to choose the top features from each role group (where the top is determined by a ranking of effect for evidence and counter evidence, and by importance for missing evidence).

3.3. Handling Priors

Many classifiers have priors for each of the classes. We consider those additional features for the purpose of justification. In Logistic Regression, each class j has an intercept which we add as a feature that has the intercept value as the importance for j, and zero importance for all other classes. The effect is always equal to the importance, so the only possible roles for the priors are normal evidence and normal counter-evidence (if they are not negligible).

3.4. Producing a Narrative

Once the key features and their roles are known, the justification narrative can be generated textually and/or visually. To make it concrete, consider a prediction made by the classifier described earlier. The predicted class is Y, and the selected features for the justification narrative are SLOPE, VENUE, DIAMETER and Y-PRIOR (the prior of class Y).

Figure 1 shows one possible way to present the justification narrative, generated by our package. In this example, we generate a short textual summary of the narrative (using simple templates), a bulleted list of the key features, and a bar chart showing the relative effect of the features.

It should be stressed that the narrative is not a particular visualization, however. It is a set of features with associated roles, effects and importance values. It can be visualized using charts, tables, bulleted templated phrases or in the form of an essay using a full Natural Language Generation system. We provide multiple such outputs as options, as the best way is ultimately application-dependent.

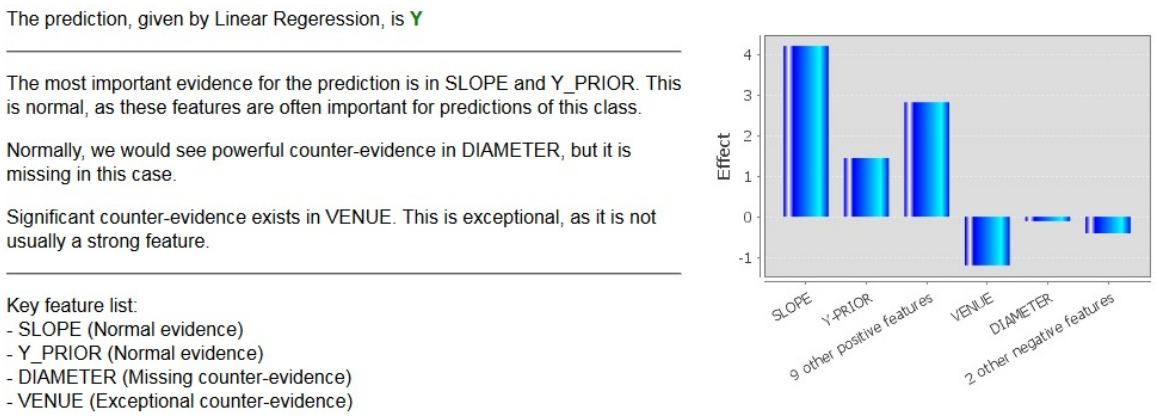

Figure 1: Justification narrative example

4. Package

PreJu is a Java package[1] that generates justifications for classifier predictions. PreJu works in one of two ways: as a stand-alone configurable tool, it accepts input in the form of XML providing the effect and importance scores of the features, leaving the implementation details to the user, and allows the configuration of key feature selection, role assignment and output types via XML. As an API, it provides simple interfaces for producing justifications programmatically and contains implementations for Weka's Logistic Regression and Linear Regression classifiers as integrated input sources.

5. Future Work

One part of future work is the continued expansion and refinement of our package. Handling additional model types is a priority, as well as providing implementations for major third-party packages. Another important task is the evaluation of our method in terms of similarity to expert-produced justifications and user satisfaction.

We are also continuing research to facilitate better justifications. In particular, the current method lacks the ability to explain the features themselves and the real-world meaning of their values (e.g., what is a network diameter? what does a large diameter entail?). We are exploring potential methods of obtaining such explanations automatically (for example, from Wikipedia or from relevant text books) using Information Extraction techniques. At the same time, we are looking at Natural Language Generation techniques for creating better, more expressive text describing the narrative and the meaning of features.

Acknowledgments

This research is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

1. Available at http://www.cs.columbia.edu/~orb/preju/

References

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11, August 2010.

Emilio Carrizosa, Belen Martin-Barragan, and Dolores Romero Morales. Detecting relevant variables and interactions for classification in support vector machines. Technical report, 2006.

Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, 2000.

C. Lacave and F. J. Díez. A review of explanation methods for Bayesian networks. *Knowledge Engineering Review*, 17:107–127, 2002.

Brian Y. Lim and Anind K. Dey. Toolkit to support intelligibility in context-aware applications. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, 2010.

M. Robnik-Sikonja and I. Kononenko. Explaining classifications for individual instances. *Knowledge and Data Engineering, IEEE Transactions on*, 20(5):589–600, May 2008.

Edward H Shortliffe and Bruce G Buchanan. A model of inexact reasoning in medicine. *Mathematical biosciences*, 23(3):351–379, 1975.

William R. Swartout. Xplain: A system for creating and explaining expert consulting programs. *Artif. Intell.*, 21(3), September 1983.

Duane Szafron, Russell Greiner, Paul Lu, David Wishart, Cam Macdonell, John Anvik, Brett Poulin, Zhiyong Lu, and Roman Eisner. Explaining naive bayes classifications. Technical report, 2003.

R. Teach and E. Shortliffe. An Analysis of Physician Attitudes Regarding Computer-Based Clinical Consultation Systems. *Computers and Biomedical Research*, 14:542–558, 1981.

Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, 2007.

Jo Vermeulen. Improving intelligibility and control in ubicomp. In *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Adjunct*, Ubicomp '10 Adjunct, 2010.

Ghim-Eng Yap, Ah-Hwee Tan, and Hwee-Hwa Pang. Explaining inferences in bayesian networks. *Applied Intelligence*, 29(3):263–278, 2008.