When to Trust Predictions from Independent Classifiers

Jonathan Brophy
Department of Computer Science
Eugene, OR 97403
jbrophy@cs.uoregon.edu

Abstract

There is no doubt that machine learning systems are rapidly finding their way into more people's lives every day, from the simple bag of words model, to the very complex architectures of various deep neural networks. As these models become increasingly complex, the ability for them to explain a prediction about a particular instance generally diminishes. This paper reveals what has been done in the area of explaining predictions, and deciding whether or not a model is trustworthy enough to be deployed into the real world. This review covers how explanations are generated in different ways from simple models like decision trees and Bayes nets, to more model specific applications like classifying images or semantic segmentation. Finally, a few general purpose frameworks are explored that attempt to explain the predictions of any classifier and ultimately decide if a model is trustworthy or not.

1 Introduction

As machine learning grows in popularity and widespread use, more and more questions arise about the validity of the predictions being produced. Many machine learning systems are treated as a black box, where the internal workings are not always completely understood. This makes the output of these systems questionable, particularly when the decisions being produced affect people's lives in significant ways. This is especially true for medical diagnoses. Medical professionals are rightfully hesitant to deploy models that are unable to clearly and intuitively explain their decisions, even if they have shown to perform more accurately than most doctors. This problem only gets worse as the complexity of the model increases, and the rate at which this is happening far outpaces the improvements in techniques and methods that try to explain these predictions.

In the deployment of a machine learning system, the learning algorithm generally tends to be a small piece in the overall process. The output of this learning algorithm can get fed into many other automated processes, effectively creating a pipeline, in which finding and debugging the machine learning system can become even more difficult (Figure 1). Just like any other piece of software, problems in the learning code can create technical debt, slowing down software development (Sculley et al. 2015). Any attempt to modularize individual components can relieve stress on the whole system. This paper focuses on the machine learning component, where the ability to explain a model's predictions can improve debugging, transparency, and trust in the machine making the predictions.

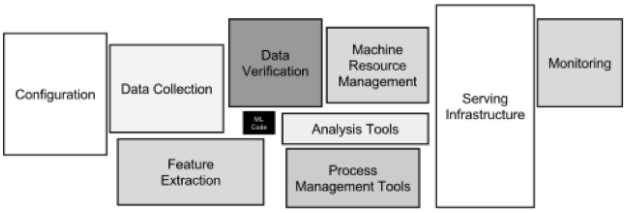

Figure 1: Sample deployment of a real-world machine learning system containing various components. The learning algorithm is the small black box near the middle (Sculley et al. 2015).

The rest of this paper discusses how some models are naturally able to explain their predictions, how other models such as vision systems have attempted their own custom solutions to explaining their decisions, and finally a few new frameworks that attempt to generalize explaining predictions for any classifier.

2 Simple Models

As mentioned earlier, learning algorithms that generate simple models tend to have an easier time explaining how they arrived at a particular output. Two models immediately come to the forefront of this discussion, decision trees and Naive Bayes.

Decision trees use the information gain from each feature to build a tree that is easily interpretable by most humans. For any new instance the model attempts to classify, a path to that classification label can easily be traced back using the model's tree. This is a big part of the appeal of using decision trees. The explanation is simple enough that any non machine learning expert can begin to understand the model's decisions.

Naive Bayes treats all features independently, vastly simplifying the model's complexity and computation. Because of this, it is easy to see what features contribute the most in any particular instance by applying the logarithm to the

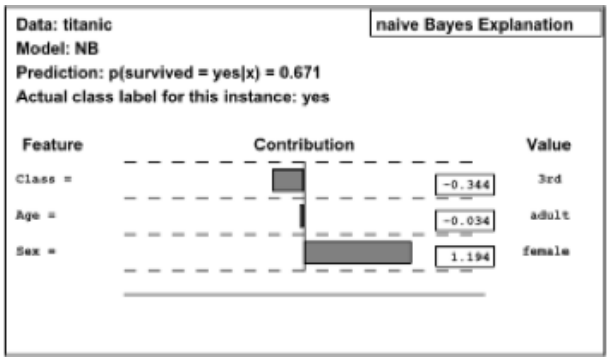

Figure 2: Explanation of an instance from the titanic dataset using a naive Bayes classifier (Kononenko and others 2010).

model's equation (Kononenko and others 2010). Each feature's relative importance can be computed and shown to favor a particular class label for any instance (Figure 2).

Decision trees and naive Bayes are useful, but these simple models are not sufficient for many tasks, leading to the adoption of more complex techniques, such as random forests, support vector machines, neural networks, etc. Neural networks are notoriously difficult to interpret, as layers of non-linearity make it harder to find out which features contribute how much. Also, the popular trend of end-to-end learning and making models deeper and deeper only exasperate this task.

One other explanation that does not quite fit under the category of a 'simple' model is matrix facotrization. Matrix factorization is not intuitively interpretable and thus a new technique created proxy models such as first order logic (FOL) and Bayes nets (BN) to explain the original model's predictions (Sanchez et al. 2015).

3 Vision Systems

There has been remarkable progress in the area of computer vision thanks to the constant emergence of new and complex learning systems. Many of these advances come from convolutional neural networks with varying deep architectures.

3.1 Image and Text Alignment

A lot of this work in the computer vision community focuses on taking an image, engineering features, and classifying the entire image. This is certainly useful, but not every image can be summed up in a few words. It is more desirable to have in-depth descriptions of what is depicted in the image. By using multi-modal data of visual images and text information, it is possible to train a model to align text to the correct regions in the image. Not only that, text descriptions can be generated for image regions by modeling language using recurrent neural networks conditioned on the image regions generated from a convolutional neural network (Karpathy and Fei-Fei 2015).

This type of model not only helps parse all of the information in an image that humans can do in glance, but breaks the image down into multiple pieces that can be analyzed individually. This can help shed some light into how a model

Figure 3: Examples of what part of the image is being focused on while generating the underlined word in each caption (Xu et al. 2015).

arrives at a final summary description of an image by considering the various image regions in the picture.

3.2 Visual Attention

In addition to generating captions for images and image regions, a model has been developed that highlights which part of the image is taken into consideration when generating a word. Annotation features are generated from a convolutional network while a long short-term memory (LSTM) network generates words one at a time based on previous words. The model's attention while generating a word in the caption can then be found by treating potential locations in the image as latent variables, and the probability of a word being generated given an image feature vector can be computed (Xu et al. 2015).

Correct examples of the model focusing on part of the image while generating words are useful in explaining what the model is doing while generating each word (Figure 3). If the model were generating a word, and the word didn't correspond to anything in the image, and we could not see what the model was focusing on, then it would be hard to tell exactly what the model was doing. But since you can see exactly what part of the image is highlighted during word generation, you get a much better idea of how the model came to the conclusion it did.

3.3 Predicting Failures

Another interesting avenue for model explanations deals with predicting potential failures. There has been work done for various image tasks such as semantic segmentation, vanishing point and camera parameter estimation, and image memorability prediction to detect if failure to produce the correct output for an image is imminent. This system, called ALERT, attempts to gauge the input image for quality, and based upon this analysis, provide insights into which features will be useful, and which can effectively be ignored (Zhang et al. 2014).

This type of system is nice because it can be used to help explain a model's predictions without messing with the inner workings of the model. This is a very convenient feature as many different models are often pipelined together, making it hard to easily insert a tool to explain the model's decisions.

4 General Explanation Methods

As described in the previous section, it can be beneficial to have a tool that can explain predictions from any classifier,

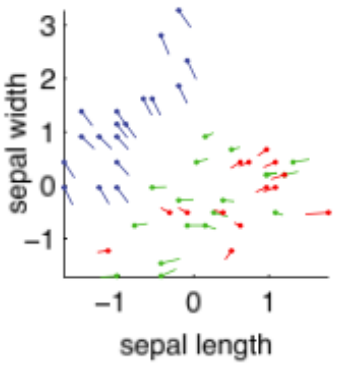

Figure 4: Example of the gradients for features sepal width and sepal length in the Iris dataset. Class labels: Iris setosa (blue), Iris virginica (red), and Iris versicolor (green) (Baehrens et al. 2010).

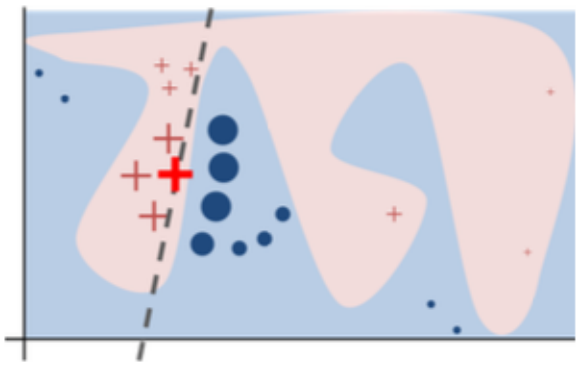

Figure 5: Example of a model m's decision boundary (blue/pink), and the instance being explained (highlighted red cross). Other instances are sampled, predicted using m, and weighted (size of the shapes). The dashed line is the local linear explanation (Ribeiro, Singh, and Guestrin 2016).

without having to dive into the details of the model. For example, say you have a model that generates predictions in a specific domain, and now you want to substitute in a completely different model to do the predictions. This will require a lot of work to build in a new explanation system for this new classifier.

Recent work focuses on the change in output based on different feature sets. To find out how much each feature contributes to a specific classification, the prediction difference is recorded, which is the difference between the expected prediction when the classifier uses all feature values versus when the classifier 'ignores' a feature value (Kononenko and others 2010). This is a reasonable start, but often times, features are correlated with one another, and the prediction of a model can change significantly depending upon which features are ignored. This requires all possible subsets of the features to be taken into account for the prediction difference. If, for example, all n features in a model are binary, this will require 2^n subsets of predictions for a particular instance, an exponential problem. To overcome this, an approximation to the actual explanation is explored by taking m samples of prediction differences, showing that useful explanations can be computed in polynomial time (Kononenko and others 2010).

Another framework uses the same basic idea of prediction differences from different feature values but uses the features' gradient to visualize how much a feature value needs to change in order to push a classification to a different label (Baehrens et al. 2010). They find the gradient of a feature against another feature to see how much change is needed to predict one class over another (Figure 4).

Explaining individual predictions is certainly a step in the right direction, but a person might reasonably be more interested in knowing if they can trust the entire model as a whole. A new method called Local Interpretable Model-agnostic Explanation (LIME) attempts to explain individual predictions using sparse linear explanations (Ribeiro, Singh, and Guestrin 2016). Linear explanations are nice in that they explain the decision for a given instance (Figure 5), but local explanations do not automatically translate to reliability for the entire model.

In addition to LIME, submodular pick (SP-LIME) attempts to pick explanations of instances that cover as many features as possible. It does this by creating a matrix of explained instances versus features and picking explanations where features are marked as important in the decision process (Ribeiro, Singh, and Guestrin 2016). Redundant explanations are avoided from being picked as the desired explanations should cover a diverse set of instances. This coverage is maximized in the SP-LIME algorithm to measure a model's trustworthiness.

5 Discussion

There is growing attention to the idea of transparency and accountability on the part of machine learning algorithms. This is due to the increasing complexity of new models and it is important that researchers take this topic seriously. Ultimately, it will not matter how accurate a model is, not many people will want to use it if there are no rational explanations behind it's decision making progress. As machine learning encroaches more into people's personal lives, a model that can explain itself becomes increasingly important.

We have seen how some simple models are able to explain their predictions just because of their model characteristics. There has also been much work done to increase the interpretability of various vision systems with multiple different techniques. Finally, some work has been done to come up with more general methods of explaining individual predictions for any type of classifier, as well as full model trustworthiness.

This is a growing topic in machine learning and more research will continue to improve the explanations of complex learning systems. I think there is still much improvement to be made for explaining deep models, as these are becoming increasingly popular. There is also room to improve explanations for probabilistic graphical models that process interconnected data. When you want to jointly reason over mul-

tiple instances, the problem then becomes figuring out how much each feature contributes to an instance as well as how much each instance contributes to other instances. It is encouraging to see that researchers are putting in more effort to tackle these problems and it is exciting to see what solutions come from these efforts.

References

Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and MÃžller, K.-R. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11(Jun):1803–1831.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3128–3137.

Kononenko, I., et al. 2010. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research* 11(Jan):1–18.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " why should i trust you?": Explaining the predictions of any classifier. *arXiv preprint arXiv:1602.04938*.

Sanchez, I.; Rocktaschel, T.; Riedel, S.; and Singh, S. 2015. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Symposium on Knowledge Representation and Reasoning*.

Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.-F.; and Dennison, D. 2015. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, 2503–2511.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2048–2057.

Zhang, P.; Wang, J.; Farhadi, A.; Hebert, M.; and Parikh, D. 2014. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3566–3573.