A Critical Review: Counterexample-Guided Data Augmentation, IJCAI 2018
Paper by: Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Kurt Kuetzer, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia
Review by: Namrata Deka

Introduction

Explosion of machine learning research has led to its application in a variety of fields. Most machine learning algorithms, especially deep neural networks, require a vast amount of data that is well distributed over the underlying feature space. Unfortunately, acquiring and annotating such data is still a tedious and costly task. In this paper, the authors propose the use of misclassified instances (counterexamples) to augment training data sets. They present the frameworks for a counterexample generator and a data structure that analyzes the properties of the generated counterexamples to understand the model's vulnerabilities. Finally, via comparative and iterative experimentations the authors provide empirical proof of the efficacy of their data augmentation approach.

Method Summary and Critique

Systematizing the synthesis of counterexamples that a trained model is unable to classify or detect is the main contribution of this paper. By including counterexamples in the training set a model can be retrained to learn features that it had previously been unable to manipulate accurately. This idea in itself is not novel. Machine learning practitioners and engineers often analyse the features of misclassified data points to improve their models and even expand their datasets. What this paper proposes is an organized framework that can generate such counterexamples with sampling methods that maximize the coverage of the counterexample distribution.

Perhaps the most crucial component of their framework is the image generator that renders synthetic images by sampling "semantic modifications" from the space of possible configurations of elements in an image. They define semantic modifications as changes that are meaningful to the domain in question, for example the position of a car in an image for a model being trained for the self-driving domain. The authors reason that by doing so they can create counterexamples that are meaningful to the domain rather than augmenting via adversarial examples which are generated by simply perturbing a few pixel values in an image. There are two problems with this reasoning:

1. Creating this space of semantic modifications requires domain knowledge and a good grasp on the kind of features that one would want their model to learn. This condition will not always hold true, and even if it does then humans will come up with modifications that are meaningful to humans. That may not transfer the same meaning to a neural network. Anthropomorphizing machine learning models is a very common mistake. This also brings us to the second problem which is,

2. Making a conscious decision to not include adversarial examples is a bad choice that could lead to potential security threats especially in high-risk domains like self-driving cars. Also, the authors seem misguided regarding the generation of adversarial examples because they are not simple perturbations of a few pixel values in an image. Synthesizing adversarial examples and defending a model against them is a huge research area in itself - one that should not be ignored

when augmenting training sets to train models for domains where, in the authors' own words, "trustworthiness is a concern".
The image generator needs to sample from the user-specified modification space and the authors have put in a good amount of time in exploring different sampling methods. Their framework is integrated with a uniform random sampler, a low-discrepancy sampler, and a cross-entropy sampler. This exploration and inclusion is done to ensure optimal coverage of the modification space which leads to faster discovery of diverse counterexamples, and increasing the diversity of features in the augmented training set. Although, not mentioned as part of their framework, the authors have also experimented with a uniform random sampler with a distance constraint to sample modifications that are at least a certain distance from each other.
To understand the model vulnerabilities and direct the sampler towards relevant modifications, the paper proposes a data structure called *Error Tables* which the authors state to be their third main contribution. This is simply a table that contains the feature values of all counterexamples encountered so far. Studying this table gives one an idea of the kind of features that the model still needs to learn. These features can also be provided as a prior to the sampler so as to sample modifications that render images containing the same features. The authors have dedicated a lot of space to the different categories of features and how the feedback to the sampler and generator is done. Most of it is quite trivial and has been overemphasized on. The key takeaway from this section is the feature analysis segment which uses principal component analysis to determine the most useful features being missed by the model. PCA is a wise choice for spaces where individual features are not necessarily independent.

Results Summary and Critique
The authors carry out three distinct sets of well thought out experiments to obtain empirical proof of the efficacy of their approach. The metrics of evaluation are the average precision and recall values of object detection in every image with an IOU threshold of 0.5. Towards their stated main contribution, which is the idea of sampling counterexamples, they perform a test comparing their approach against *imgaug* - a standard image augmentation method implemented by a Python library. The results are compared against the precision and recall values obtained from each type of sampling method, with low-discrepancy Halton sampling giving the best overall improvement in accuracy due to its ability to cover the modification distribution better and provide the most diverse set of samples than the other sampling methods. Counterexample with sampling outperforms *imgaug* on the original test set as well as on all counterexample test sets.
The second set of experiments are designed to show the importance and efficiency of *Error Tables* to analyze the counterexample features and direct the sampler towards a narrower subset of modifications to choose from. The number of augmenting images added to the training set to get results comparable with the previous experiment was only a third of the number of augmenting images required previously. This shows that with intelligent sampling that ensures a good coverage of features to learn from, fewer data and therefore lesser time is required to retrain the model.
Finally, to understand their model's behavior after being re-trained with counterexamples the authors iteratively sample batches of counterexamples and re-train their model until they reach a saturation point where the generator is unable to find any more counterexamples. It is observed that the accuracy on both original and counterexample test sets improve with every cycle of

augmentation - suggesting that the model does not overfit to counterexamples. Also, as iterations progress it becomes increasingly difficult to find counterexamples. Although the authors see this growing computational hardness as empirical proof of increasing assurance in the model, a concrete proof showing the relationship between the increasing time complexity, increasing model accuracy, and decreasing space of available counterexamples would have given a generalized picture of the robustness of this method.

Possible Extensions

This work could be improved further or extended by exploring the following:

1. Can we use analytical machine learning and statistical tools to learn the modification space instead of manually designing the semantics? This should, in theory, construct a modification space that is more relevant to the characteristics of the model in use. It could be interesting to examine what semantics it ends up learning and how the performance compares against the current design.
2. Inclusion of adversarial examples. As mentioned earlier, counterexamples should ideally be inclusive of adversarial examples as well. To keep the domain relevance intact, one could always generate targeted adversarial examples and include them in the augmented set. Further work can be undertaken to see its effect on resulting accuracies, especially on precision values.
3. This work could be extended to the setting of unsupervised learning. How does one define, represent and determine counterexamples in unlabelled data? How can we augment a dataset being used to learn an autoencoder for instance? Can we improve the feature approximation that the encoder represents?
4. Is it possible to get proofs of convergence and theoretical bounds on augmentation loops and their time complexity? How will this vary across models of varying complexity given a fixed size and distribution of initial data to begin with?

Potential Applications

1. The proposed framework can be applied in any supervised setting where well distributed annotated data is difficult to come by, continuous testing in the real world is a costly affair, and the risk of misclassification is high. For example, image processing in self-driving cars as we have seen in the paper, detecting/tracking wildlife and poachers in parks and sanctuaries, surveillance systems to detect suspicious activities, etc.
2. The sampler and generator can be used to test the performance of a trained model. Their inability to generate counterexamples can be taken as an empirical assurance that the model under consideration has learned the relevant space of salient features well. This can be added as an extra measure along with evaluating validation accuracies during a testing pipeline.
3. With further investigation and formulation this framework could potentially be applied to inverse reinforcement learning (IRL) tasks. Expert demonstrators in IRL only demonstrate a handful of optimal trajectories that usually do not cover the entire state-action space. Using the feature analysis, sampling and generation techniques proposed in this work one could generate more optimal trajectories to train the policy and reward models on.