Anomaly detection in image data sets using deep learning

Kumar Saketh, Ali Radha
Department of EECS
University of Michigan
{ksreddy,aliradha}@umich.edu

1. Introduction

Anomaly detection is the task of identifying entities in a data set which differ from the nominal majority. Anomaly detection has been studied extensively for structured multivariate data where each entity is represented by a set of d features. Recently, there has been some work on solving anomaly detection for unstructured data such as time-series, network graphs and images [1] . However, most of these methods rely on extracting hand-engineered features from the unstructured data and subsequently applying anomaly detection methods for structured data.

With the advent of deep learning, feature representations can now be learned automatically from large unlabeled collections of images. It has been shown that the features inherently learned by these deep architectures have been successful in several tasks in computer vision including classification and segmentation. We are keen to build off of this work and use the features learned from deep neural networks to detect anomalous images. If we are successful, our proposed idea will have the advantage over traditional hand-engineering based methods in that (i) the algorithm will generalize to a wide variety of image data sets, and (ii) the performance of our algorithm, as is the case in classification tasks, could potentially be better that the performance of hand-engineered systems.

There are several algorithms for detecting anomalous images using hand-engineered features [1]. Currently, the only algorithm we know of for detecting anomalies using deep networks is based on identifying images with large reconstruction errors [2]. The basic idea in this work is to first build an auto-encoder to learn features, and then subsequently calculate the reconstruction error for each image in the data set, and declare images with large reconstruction errors as anomalies. This method, henceforth referred to as RE, was applied to the MNIST data set in [2] to detect 'ugly' hand-written digits.

RE will work well for clean image data sets such as MNIST where each image has only one object of interest (a digit in the case of MNIST), and there is no background clutter. However, for data sets where there is background noise, RE can fail in the following scenario - for images with nominal objects but unique background clutter, the reconstruction error can be high because of the auto-encoder model's inability to reconstruct the background. As a result, these images will be classified as anomalies, resulting in false alarms.

For instance, consider a toy data set where a majority of the images are of cats with no background. Also assume that this toy data set has a few images of cats with background, and also a few images of dogs, like in Figure 1. RE would then classify images of dogs as anomalies as expected, but additionally, they would also classify images of cats with background as anomalies because the model will fail to reconstruct the background. Instead, we would like to design algorithms which will detect only the dog images as anomalies.

2. Proposed work

To overcome the drawback of the RE method, we propose the following. First, we make the observation that the features learned in a deep network are invariant to confounding image attributes [3] and therefore inherently filter out background clutter. We plan to exploit this property by detecting anomalies based on the features learned by the network as opposed to detecting anomalies based on the reconstruction error. Next, we formally describe our proposed approach.

2.1. Framework and Approaches

Assume a data set of unlabeled images $\mathcal{I}$ comprising of n images $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$. The goal is to detect anomalous images from $\mathcal{I}$. Our proposed algorithm has the following three steps: (Step 1) Build a unsupervised deep network $\mathcal{D}$ to model $\mathcal{I}$. The model can either be based on auto-encoders or probabilistic networks such as deep belief networks. (Step 2) Use the learned network to derive features for anomaly detection by adopting one of the following two alternatives A1 or A2: (A1) Feed each image I_j through $\mathcal{D}$, and obtain the intermediate features F_j - the middle layer of auto-encoders, or the final layer of deep be-

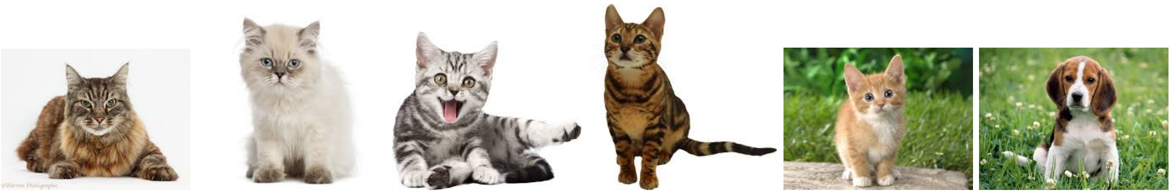

Figure 1. First four images are nominal cat images with no background, next image is of cat with background, last image is of dog.

lief nets - with respect to $\mathcal{D}$ corresponding to I_j. (A2) Feed each image I_j through $\mathcal{D}$, and identify the subset of nodes S_j in $\mathcal{D}$ that are activated by I_j. If $\mathcal{D}$ has k hidden nodes, we represent each S_j as a sparse, binary k-dimensional vector with the non-zero entries in S_j corresponding to the node indices of the nodes activated in $\mathcal{D}$ by I_j. (Step 3) Detect anomalies based on the features $\mathcal{F} = \{F_1, F_2, \ldots F_n\}$, or $\mathcal{S} = \{S_1, S_2, \ldots S_n\}$, by using an anomaly detection algorithm for multivariate data. We plan to use the iForest algorithm [4], which is state-of-the-art, for this purpose.

In contrast with RE, we expect our proposed method to detect only images which are different with respect to the features learned by the deep network as anomalies, and not images with nominal objects of interest, but with unique background clutter. For instance, in the toy example, we expect our method to only detect dog images as anomalies.

2.2. Software, Data Sets and Things to Learn

We are planning to use the Theano package for implementing deep networks.

First, we will start off by detecting ugly digits in the MNIST data set, which has images with clean backgrounds. When working with MNIST, we will compare the anomalies generated by our method with the anomalies generated by the RE method. For this data set, we expect the results generated by our method and the reconstruction error based method to be similar. Next, we will work with the ImageNet data set. From the ImageNet data set, we will create subsets for anomaly detection task by including all images in the ImageNet data set from certain classes (for example: sky, flowers, trees), and then contaminating this data set by introducing a few images of certain other classes (for example: dogs, cats). We will not use the labels while running the algorithm, but will use the labels to evaluate the performance of our algorithm and contrast this performance with the RE based method.

To work on this project, we will need to learn about existing work on deep networks, and also on how to use Theano in order to implement them. We will also need to learn and implement the iForest algorithm.

3. Expected outcome

If successful, we expect to have designed an automatic algorithm that is invariant to background clutter for detecting anomalies in image data sets. We will base our evaluation of whether we are successful on the MNIST data set by (i) visually eyeballing the results, and (ii) comparing our results with RE which has been successful in identifying badly written digits. Next, we will base our evaluation of success on the ImageNet data set by utilizing the labels available in the data set. We would consider our proposed method to be a success if we obtain a high value and outperform RE with respect to the area under the ROC curve.

The components that we think are risky are the following: (i) Training deep networks is tricky [5]. (ii) The features learned from the deep networks, or the node activation patterns, could be fairly high-dimensional, which could be hard because of the curse of dimensionality. (iii) Our hypothesis that our method will work better than the RE method could potentially fail to hold in practice, in which case we might not get a significant improvement over RE.

We are confident that we will be able to build auto-encoders by following the tutorial by Andrew Ng [6]. Therefore, the least we expect to deliver, subject to the risky items failing, is reproducing the RE method [2] and its results on MNIST. If we are successful in training DBN's, we will then have an alternative deep network structure to the auto-encoders off which we can detect anomalies by comparing the likelihood instead of the reconstruction error as is the case with RE. If we are successful in getting the anomaly detection method to work, either by limiting the dimension of the feature space $\mathcal{F}$ or by exploiting the sparse nature of $\mathcal{S}$, we should have a working algorithm that should at the very least work as well as RE, if not better.

References

[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. ”Anomaly detection: A survey.” ACM Computing Surveys (CSUR) 41.3 (2009): 15.

[2] `http://learn.h2o.ai/content/hands-on_training/anomaly_detection.html`

[3] Goodfellow, Ian, et al. ”Measuring invariances in deep networks.” NIPS. 2009.

[4] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. ”Isolation forest.” Data Mining, 2008. ICDM'08.

[5] Hinton, Geoffrey. ”A practical guide to training restricted Boltzmann machines.” Momentum 9.1 (2010): 926.

[6] Ng, Andrew, et al. ”UFLDL tutorial.” (2012).