Regularized likelihoods

Author(s): Ioan-Mihail Dinu
Laboratoire de Phisique de Clermont, Clermont-Ferrand, France

0.1 Method

The method presented in this paper attempts to use the power of generative models for the downstream task of Anomaly Detection. We have mainly explored the possible applications of flow-based methods, since they have the advantage of providing an explicit likelihood.

Normalizing Flows (NF) are one of the best methods available at the moment for density estimation in high-dimensional data (Ref. [1]). Those types of models work by learning a bijective mapping between the data distribution and a multivariate gaussian (with the same number of dimensions). Experience shows that, unfortunately, the likelihood that NF models provide is not sufficient as a stand-alone anomaly detection metric.

In an attempt to *regularize* the likelihood obtained with such density estimation techniques we have explored several alternatives to the vanilla NF models. One particularly interesting approach is the $\mathcal{M}$-flow model introduced originally in Ref. [2].

0.1.1 $\mathcal{M}$-flows

The $\mathcal{M}$-flow model combines the idea of reconstruction error from autoencoders with the tractable density of NF. If there exists a lower-dimensional data manifold embedded in the data space, this method attempts to learn both the shape of this data manifold $\mathcal{M}$ and the density over that manifold.

In order to create a $\mathcal{M}$-flow we start with a bijective mapping f between the latent space $\mathrm{U} \times \mathrm{V}$ to the data space X, as in Eq. 0.1. The latent space is split in two components: $\mathbf{u}$, which is the latent space representation that maps to the learned manifold, and $\mathbf{v}$, which represents the remaining latent variables that are "off the manifold".

$$\begin{aligned} \mathrm{f} : \mathrm{U} \times \mathrm{V} &\to \mathrm{X} \\ u, v &\to \mathrm{f}(u, v) \end{aligned} \tag{0.1}$$

The transition from the space $\mathrm{U} \times \mathrm{V}$ space to the space U is implemented as a projection operation, the $\mathbf{v}$ component being basically discarded. The inverse of this transition is implemented with zero-padding, $\mathbf{u}$ remains unchanged and $\mathbf{v}$ is filled with zeros. We notate the previous operations with the function g, characterizing the transformation of a latent representation $\mathbf{u}$ to a data point $\mathbf{x}$ (shown in Eq. 0.2).

$$\begin{aligned} \mathrm{g} : \mathrm{U} &\to \mathcal{M} \subset \mathrm{X} \\ u &\to \mathrm{g}(u) = \mathrm{f}(u, 0) \end{aligned} \tag{0.2}$$

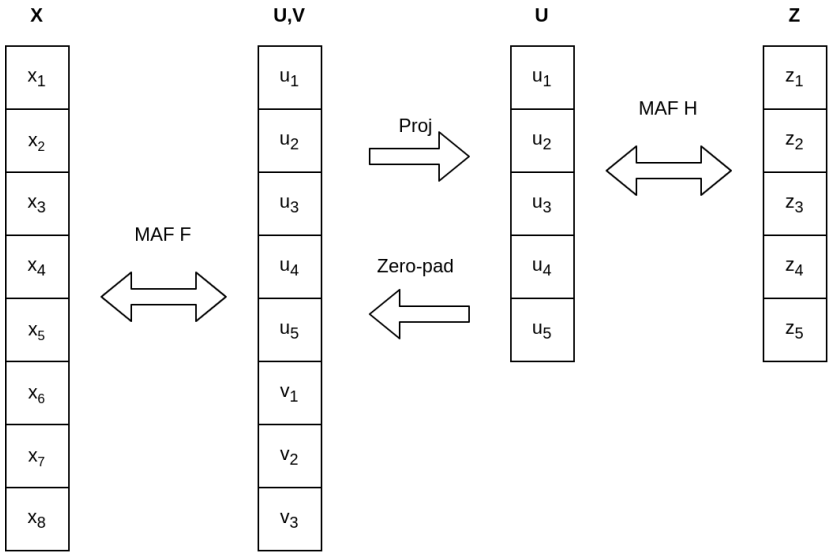

Figure 1. An example representation of dependencies between the data **x**, latent variables **u**, **v** and the normally distributed variable **z**. Here the example data has 8 dimensions and the latent space has 5 dimensions. The bijective transformations are learned with Masked Autoregressive Flows (MAFs).

Finally the density in the space U is learned using a regular NF model denoted as h. A schematic representation of those operations is presented in Fig. 1.

The training of this model is split in two phases completed sequentially for every batch. Firstly, the parameters of f are updated by minimizing reconstruction error from the projection onto the manifold (loss function in Eq. 0.3). The second phase of training consists in updating the parameters of h by minimizing the negative log likelihood from Eq. 0.4.

$$\mathcal{L}_{manifold} = ||x - g(g^{-1}(x))||^2 \tag{0.3}$$

$$\mathcal{L}_{density} = \log p_u(g^{-1}(x)) \tag{0.4}$$

Regarding the preprocessing steps, the LHC Olympics datasets have been clustered and the the following features have been selected for each of the two leading jets: p_T, η, E, m, τ_3/τ_2, τ_2/τ_1, where τ_n is the n-subjettiness. For these 12 features, the best performing manifold size was 8.

This model offers the possibility to calculate both the density on the manifold and the reconstruction error from the projection on the manifold. We tried to use both of those metrics in order construct a robust anomaly score as in Eq. 0.5. This metric performs the anomaly detection task better on the R&D dataset that its components and better than a basic normalizing flow model trained on the same data, judging by the ROC curves in Fig. 2.

$$\mathcal{R}_{exp}(x) = \frac{||x - g(g^{-1}(x))||^2}{1 + p_u(g^{-1}(x))} \tag{0.5}$$

While experimenting with this anomaly score, it became apparent that it generates a bias towards events with high dijet mass (m_{jj}). In order to decouple $\mathcal{R}_{exp}$ from m_{jj} we included the marginal likelihood of m_{jj}, that was modeled using Kernel Density Estimation (KDE), as a term into the anomaly score. The resulting metric, denoted $R_{m_{jj}}$, uses the ratio between the likelihood on the manifold and marginal m_{jj} likelihood as in Eq. 0.6.

$$\mathcal{R}_{m_{jj}}(x) = \frac{||x - g(g^{-1}(x))||^2}{1 + \frac{p_u(g^{-1}(x))}{p_{KDE}(m^x_{jj})}} \tag{0.6}$$

Translating the performance obtained on the R&D data to the black boxes proved to be a big challenge. The small differences in modeling from a black box to another are often enough to introduce significant biases. The only apparent solution seems to be training and applying the method on the same dataset.

0.2 Results on LHC Olympics

The R&D dataset was heavily used for benchmarking different approaches, Fig. 2 shows the anomaly detection performance of different metrics on the R&D dataset.

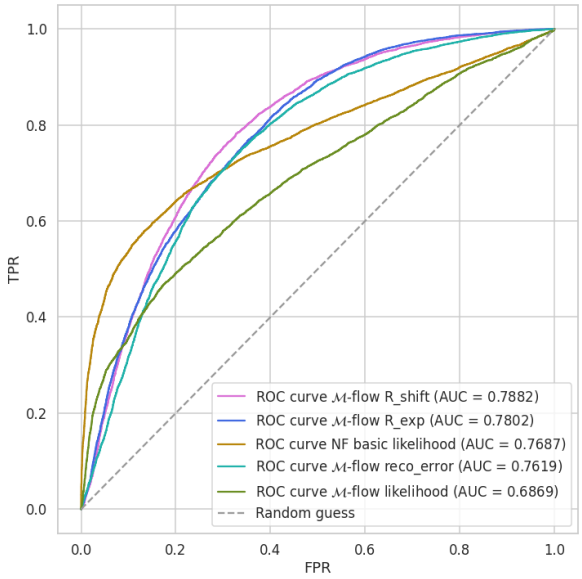

Figure 2. Signal detection ROC curves in the R&D dataset for different anomaly scores

In order to evaluate the performance of this method in the absence of pure background training data, a small fraction ($\sim 1\%$) of signal was introduced into a subsample from the R&D dataset. The resulting data sample was used both for training and evaluation of the model.

Several cuts have been applied on $\mathcal{R}_{m_{jj}}$ while trying to find any indication of a resonance in the m_{jj} spectrum. Although less apparent, there is still a bias towards identifying higher

m_{jj} events as being anomalous. The right plot in Fig. 0.2 shows the m_{jj} distribution for events above the 50^{th} percentile of $\mathcal{R}_{m_{jj}}$ vs events above the 70^{th} percentile of $\mathcal{R}_{m_{jj}}$. If we were to take the 50^{th} cut as a baseline, it is clear that increasing the threshold has the effect of selecting events with slightly higher m_{jj}. Unfortunately there is no sharp peak in the m_{jj} distribution that would indicate a possible resonance, but rather the tail of the distribution seems to get bigger.

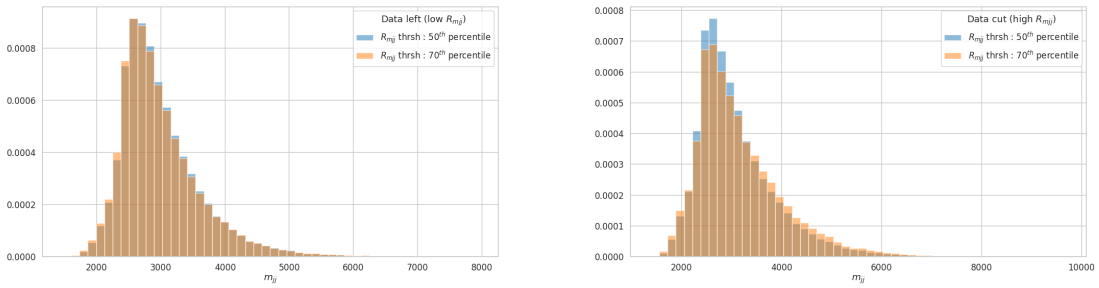

Figure 3. Overlapping m_{jj} distributions below (left) and above (right) two threshold cuts on $\mathcal{R}_{m_{jj}}$. Distributions for a 50^{th} percentile cut are in blue, while distributions for a 70^{th} percentile cut are in orange. The x axis is in GeV/c^2.

The results so far suggest that this method can not be used reliably to find the hidden signal within the black-boxes. This behavior is consistent regardless of the choice of $\mathcal{R}_{m_{jj}}$ thresholds.

0.3 Lessons Learned

One of the main lessons learned during this challenge is that: in absence of a good background model, the neural networks by themselves can not achieve good anomaly detection performance.

For the winter LHC Olympics, we approached the problem with a simple autoencoder that was trained on the full background black box. Applying that model on Black Box 1 (BB1) introduced a lot of bias that ended up acting like a fake signal. Special precautions should always be taken in order to avoid this scenario.

With the experience gained from studying BB1 we were a lot more careful to avoid creating fake signal. The subsequent problem proved to be the lack of a good background model. Since we could not relay on the full background black box, the alternative was to train on data, but this comes with its own issues.

All of the attempts so far came short of providing a good background modeling and therefore the current anomaly detection performance leaves a lot to be desired. Those trials taught us that a good machine learning anomaly detection algorithm is not just about the neural network itself, but many other analysis details should be treated with the same amount of attention.

0.4 Code Availability

Most of the machine learning heavy lifting was done with the help of the existing code base from the original $\mathcal{M}$-flow model introduced in Ref. [2] by Johann Brehmer and Kyle Cranmer. https://github.com/johannbrehmer/manifold-flow

Acknowledgments

This work was supported by the U.S. Department of Energy, Office of Science under contract DE-AC02-05CH11231.

References

[1] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1530–1538, PMLR, 07–09 Jul, 2015.

[2] J. Brehmer and K. Cranmer, *Flows for simultaneous manifold learning and density estimation*, arXiv:2003.13913.