Evaluating an Anomaly Detection System: May 30, 2021

Assume we have labelled data of anomalous and non-anomalous examples ($y=0$ normal, $y=1$ anomalous).

Training Set: $x^{(1)}, x^{(2)} \dots x^{(m)}$

↳ All normal examples, no anomalies

Cross Validation Set: $(x_{cv}^{(1)}, y_{cv}^{(1)}) \dots (x_{cv}^{(m)}, y_{cv}^{(m)})$

Test Set: $(x_{test}^{(1)}, y_{test}^{(1)}) \dots (x_{test}^{(m)}, y_{test}^{(m)})$

e.g. 10000 normal engines, 20 flawed engines

Training Set: 6000 normal engines ($y=0$)

CV Set: 2000 normal ($y=0$), 10 anomalous ($y=1$)

Test Set: 2000 normal ($y=0$), 10 anomalous ($y=1$)

60/20/20 split

Algorithm Evaluation:

Fit model $p(x)$ on training set $\{x^{(1)} \dots x^{(m)}\}$

On CV/test example x, predict:

$$y = \begin{cases} 1 \text{ if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 \text{ if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

Possible Evaluation Metrics:

- Can't use classification accuracy due to skewed data
  - → true positive, false positive, false negative, true negative
  - → precision, recall
  - → F1 score
- You can also use the CV set to choose ε
  - ↳ i.e. Whichever ε gives the highest score/accuracy

Anomaly Detection

- → very small number of positive examples ($y=1$)
- → large number of negative examples ($y=0$)
- → many different types of anomalies
  - ↳ future anomalies may look nothing like the training examples

e.g.

- → fraud detection
- → manufacturing
- → monitoring machines in a data centre

Supervised Learning

- → large number of both positive and negative examples
- → enough positive examples for model to recognize future positive examples as being similar

e.g.

- → email spam classification
- → weather prediction (sunny, cloudy)
- → cancer classification

Choosing What Features to Use!

We want $p(x)$ to be large for normal examples and small for anomalous examples.

What we don't want is $p(x)$ being comparable for both.

- Choose features that might take on unusually large or small values in the event of an anomaly.

e.g.

x_1 = memory use of computer

x_2 = # of disk accesses/sec

x_3 = CPU load

x_4 = network traffic

$$x_5 = \frac{\text{CPU load}}{\text{network traffic}}$$