**Problem Motivation**

Dataset: {x1, x2, x3, …, xm}(assume that the dataset is normal example)

Is xtest anormalous?

Usually, trend to build a model p(x). After that, check

p(x-test) < ε → anormaly

p(x-test) >= ε → OK

Gaussian Distribution

**Parameter Estimation**

X~N(μ, σ2)

μ = 1/m \* sum(xi)(i = 1 up to m)

σ2  = 1/m \* sum((xi – μ)2)(i = 1 up to m)

**Anormaly Detection Algorithm**

1.Choose features xi that you think might be indicative of anomalous examples.

2.Fit parameters μ\_1, μ\_2, …, μ\_n, σ2\_1, σ2\_2, …, σ2\_n

μ\_j = 1/m \* sum(xi\_j)(i = 1 up to m)

note: μ\_j is the average value of the j feature

σ2\_j = 1/m \* sum((xi\_j – μ\_j)2)(i = 1 up to m)

3.Given new example x, compute p(x):

p(x) = ∏nj=1 p(x\_j;μ\_j, σ2\_j)(x\_j~N(μ\_j, σ2\_j))

Anormaly if p(x) < ε

**Algorithm Evaluation**

Fit model p(x) on training set{x1, x2, …, xm}

On cross validation/test example x, predict

y = 1 if p(x) < ε(anomaly)

= 0 if p(x) >=ε(nomaly)

Possible evaluation metrics:

True positive, false positive, false negative, true negative

Precision/Recall

F1-score(the higher value this algorithm get, the better performance of this algorithm)

Can also use cross validation set to choose parameterε.(choose differentεto maximize F1-score)

**When should choose Supervised Learning and Anomaly Detection**

Anomaly Detection:

1. Very small number of positive examples(y=1)(0-20 is common).Large number of negative examples(y=0).
2. Many different “types” of anomalies. Hard for any algorithm to learn from positive examples what the anomalies looks like; future anomalies may look nothing like any of the anomalies examples we’ve seen so far.

Supervised learning:

1. Large number of positive and negative examples.
2. Enough positive examples for algorithm to get a sense of what positive examples are like, future positive examples likely to be similar to ones in training set.

Choosing what features to use

If the examples looks not like a Gaussian distribution:

You can use log(xi + c)/x1/2i/x1/3i to transform the training set looks like more Gaussian.

在异常事件中，选择那些属性值异常大或者异常小的的特征（用这些特征生成新的特征）

**Multivariate Gaussian distribution**

X Rn. Don’t model p(x1), p(x2), …, etc. separately.

Model p(x) all in one go.

Parameters: Rn, Rnxn(covariance matrix)

P(x,; ) = \* (1)

**Parameter estimate:**

Given training set:{x1, x2, …, xm}

= 1/m \* (2)

= 1/m \* (3)

**Anomaly Detection with Multivariate Gaussian**

1. Fit model p(x) by setting equation (1) and (2).
2. Given a new example x, compute expression (1) flag an anomaly if p(x) < ε.

How to choose original model and multivariate Gaussian

Original Model:

The original model probably used more often.

1. Manually create features to capture anomalies where x1, x2 take unusual combinations of values.(e.g. x3 = x1/x2)
2. Computationally cheaper(alternatively, scales better large n)
3. OK even if m(training set size) is small.

Multivariate Gaussian:

1. Automatically captures correlations between features.
2. Computationally more expensive.
3. Must have m > n, or else non-invertible.