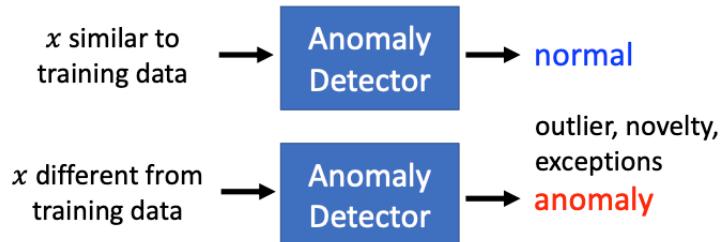


Problem Formulation

对于给定的训练数据集，做异常检测的目的是找到一个函数，这个函数可以检测输入的 x 是不是和训练数据集相似的。异常检测知识检测出和训练数据不一样的输入，检测出来的数据并不一定都是不好的，有可能是好的，也有可能是不好的。

- Given a set of training data $\{x^1, x^2, \dots, x^N\}$
- We want to find a function detecting input x is similar to training data or not.



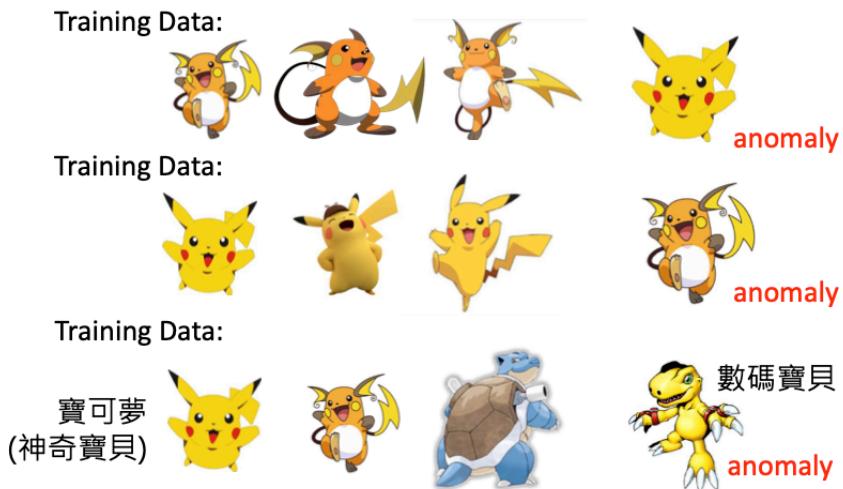
Different approaches use different ways to determine the similarity.

那么到底怎么来定义这个similarity呢？这也就是异常检测需要探讨的问题，不同的方法就有不同的定义方式。

What is Anomaly?

Q：那么我们到底如何来定义anomaly呢？如何确定我们检测出来的输入是anomaly的呢？

A：取决于具体的训练数据集。在下图中，如果训练集中有很多雷丘，那么皮卡丘就是异常；但如果有很多只皮卡丘，那么雷丘就是异常，....



Applications

异常检测有很多应用，以下简要叙述三个：

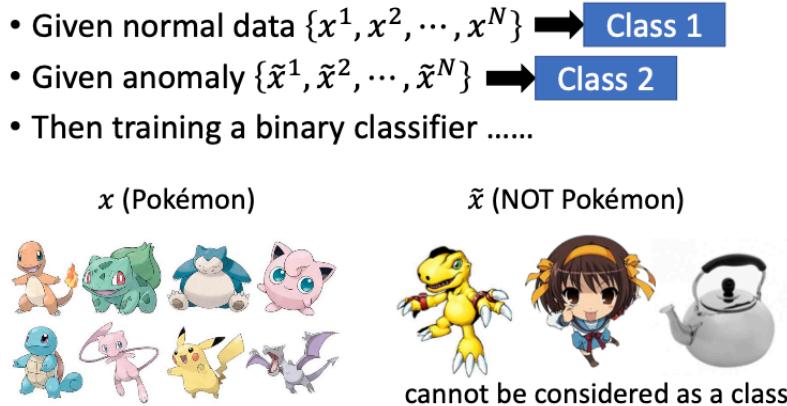
1. Fraud Detection, training data是正常刷卡行为，那么现有一笔新的交易记录 x ，我们就可以进行异常检测，来判断 x 是不是盗刷行为；比如日常都是小额消费，但突然多了一些连续的高额消费，就可以认为这是盗刷行为；

2. Network Intrusion Detection, training data是正常连接行为，现在有一个新的连接x进来，那么我们就可以让机器自己进行异常检测，来判断新连接是不是异常的；
3. Cancer Detection, training data是正常细胞的资料，比如细胞核的大小、分裂的频率等，如果来一个新的细胞x，机器可以自己决定到底是正常细胞还是癌细胞。

Binary Classification?

Q: 给定正常的数据 (Class 1) 为 $\{x^1, x^2, \dots, x^N\}$, 异常的数据 (Class 2) 为 $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^N\}$; 那么异常检测可以认为是一个二分类问题吗?

A: 如果正常数据集是“宝可梦”，那么异常数据就“不是宝可梦”，但“不是宝可梦”包含很多种类别，比如茶壶、树等；而且异常数据并不像正常数据那么容易收集；因此并不能简单地进行一个二分类问题。

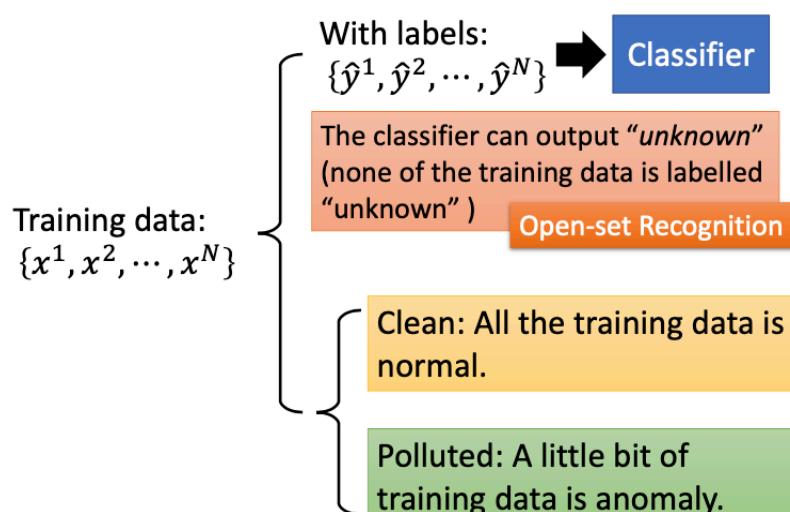


Even worse, in some cases, it is difficult to find anomaly example

Categories

异常检测可以分为以下几个类别：

1. training data带label，每个训练数据 x^i 都有对应的标签 \hat{y}^i ，一共有N个类别，那么我们就可以训练一个classifier，来进行分类，如果机器能辨别出这个input是1-N个类别，那么就认为是正常数据；如果机器不能辨别，即输出类别为“unknown”，我们就认为这是异常数据；
2. 如果training data是unlabel，这时又分为两种情况： (1) training data是clean的，所有的数据都是正常的； (2) training data中有一小部分数据是异常的，比如银行要对用户数据进行异常检测，那么training data就不可避免地包括一些异常数据。



Case 1: With Classifier

Example Application

现在我们判断一个人物是不是来自辛普森的家庭。

Example Application

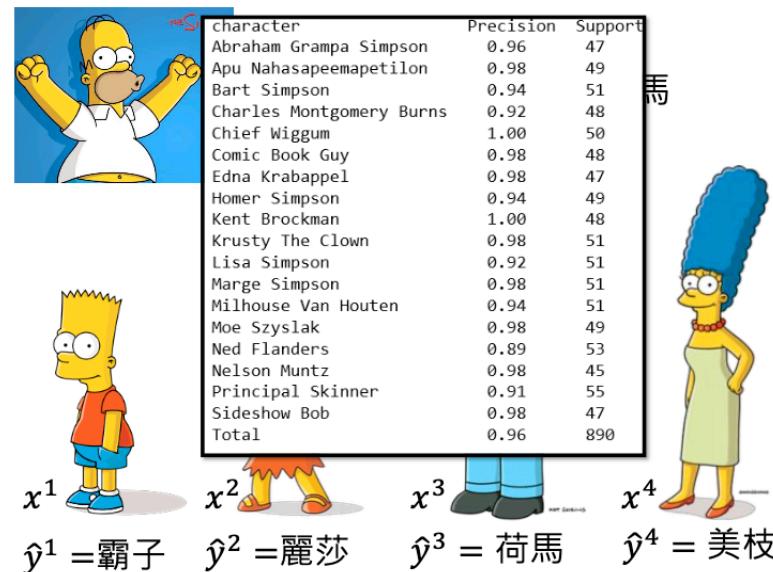
- From The Simpsons or not



x



在下图中对于输入的正常数据 $\{x^1, x^2, \dots, x^N\}$, 都有其对应的类别标签 $\{\hat{y}^1, \hat{y}^2, \dots, \hat{y}^N\}$, 那么我们就可以训练一个classifier, 来预测其输出的类别。一位很喜爱辛普森的学者进行了实验, 得出了很不错的分类结果。



How to use the Classifier

那么根据训练出来的classifier来做异常检测, 来判断一个人物到底是不是来自辛普森家庭。

我们可以让classifier再输出一个confidence score, 同时我们还需要定义一个界限threshold λ , 如果信心分数 $c(x) > \lambda$, 就认为是正常数据; 否则就认为是异常数据。



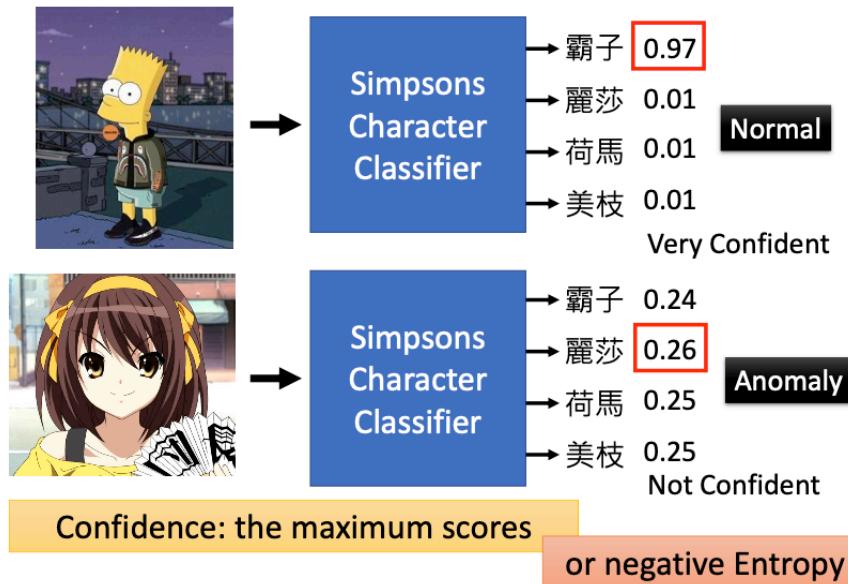
Anomaly Detection:

$$f(x) = \begin{cases} \text{normal}, & c(x) > \lambda \\ \text{anomaly}, & c(x) \leq \lambda \end{cases}$$

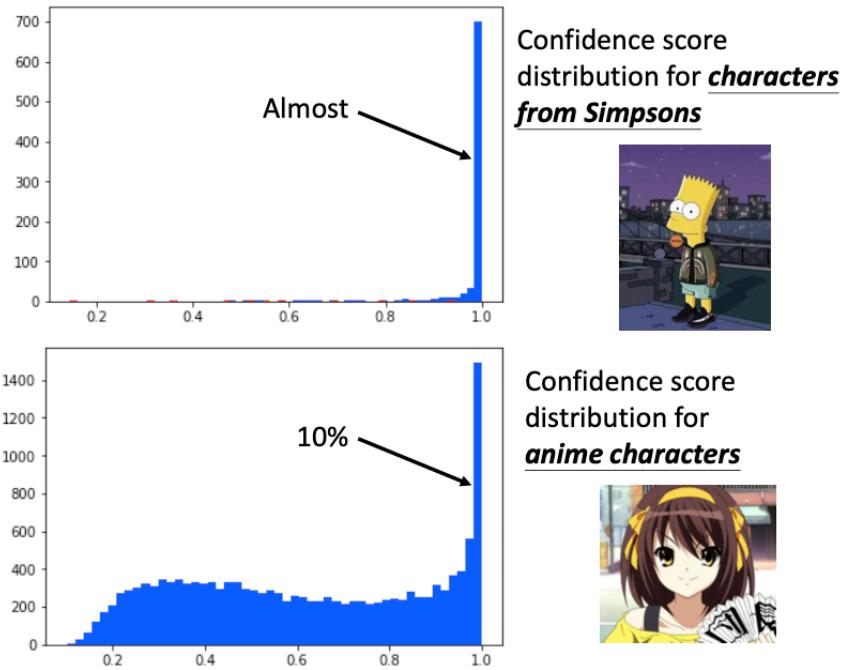
那么我们如何得到这个confidence score呢？

在下图的例子中，如果是一个辛普森家庭的人物输入classifier，这个分类器能够很自信地得出分类结果是“霸子”(0.97)；但如果是一个非辛普森的人物输入，classifier的输出分类结果就比较均匀，4个分类就都是0.20+。那么我们就可以认为classifier对第一个input是很自信的，是normal；第二个input则不自信，被认为是anomaly data。

那么我们就可以把这个confidence score认为是分类结果的distribution中的最大值，也可以做cross-entropy。



那么现在我们把辛普森家庭和其他动漫图片的所有数据都进行confidence score计算。在下图中，我们可以发现属于辛普森的confidence score distribution，几乎全都是1；但对于其他动漫的人物，只有10%的score为1，其他大多数的图片得到的分数都是比较低的。



Example Framework

训练数据集是带label的，全都是属于辛普森家庭，那么我们可以训练出一个classifier，根据这个分类器，我们可以计算出每张图片的confidence score，根据threshold来判断这张图片是不是anomaly。

Training Set: Images x of characters from **Simpsons**.

Each image x is labelled by its characters \hat{y} .

Train a classifier, and we can obtain confidence score $c(x)$ from the classifier.

$$f(x) = \begin{cases} \text{normal}, & c(x) > \lambda \\ \text{anomaly}, & c(x) \leq \lambda \end{cases}$$

Dev Set: Images x

Label each image x is from **Simpsons** or not.

We can compute the **performance** of $f(x)$

Using dev set to determine λ and other hyperparameters.

Testing Set: Images x → from **Simpsons** or not

development set中需要包括带label（是否属于辛普森）的图片，但这个图片不仅包括辛普森家庭的人物，还包括不属于辛普森家庭的人物。将数据输入训练好的classifier，根据不同的 λ ，得到系统的performance，那么我们就可以通过development set来调整 λ 的值，选择使系统得到最好performance的 λ 。

Evaluation

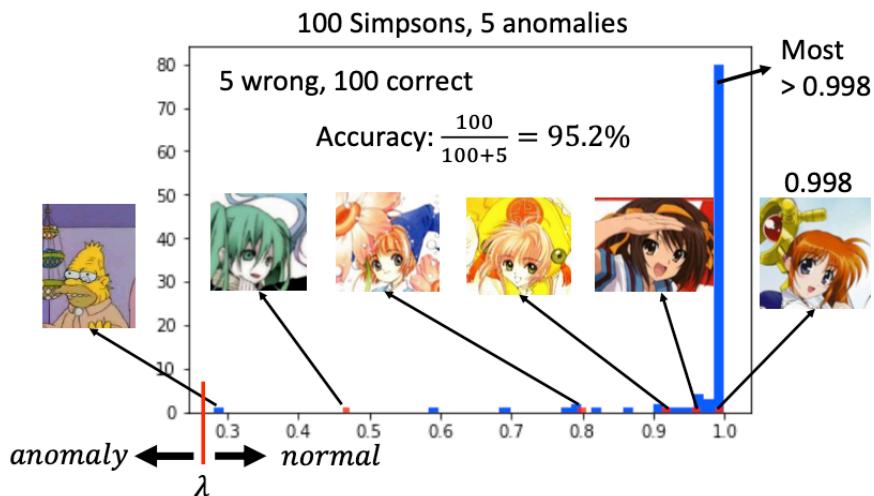
现在有100张辛普森人物的图片，5张非辛普森。在图中红色部分则表示非辛普森的score，一共有5个方块，得到的分数都很低。

正确率并不能来准确衡量一个系统的好坏，很可能一个系统有一个很高的正确率，但其实并不好。在下图中，我们可以计算出这个系统所计算出的accuracy为95.2%，但这并不是因为我们的异常检测系统很好，而是因为数据集的分布差异，异常数据所占的比例非常小。这并不是我们想要的异常检测系统，这个系统几乎会把所有的输入都判断成是异常的，但由于异常数据占比很小，最后的正确率还是会很高。

Accuracy is not a good measurement!

Evaluation

A system can have high accuracy, but do nothing.

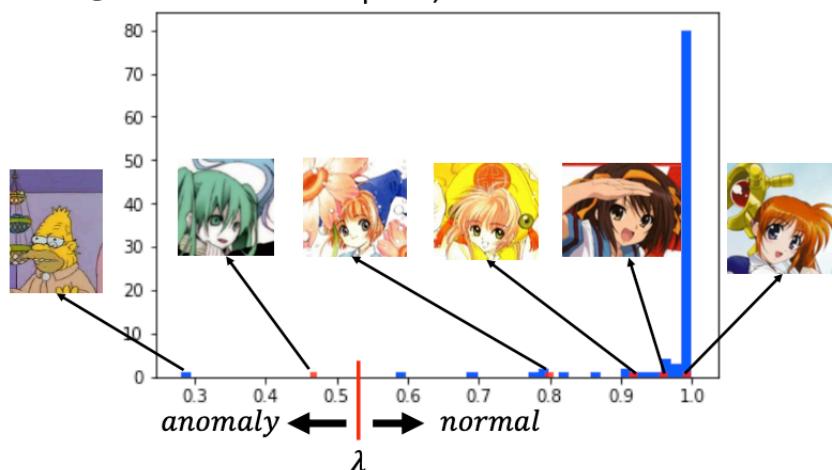


因此我们不能用正确率accuracy来判断系统的performance。

在下图中，我们把threshold设置为大于0.5的值，这时只有1个异常值被检测出来，还有剩下4个异常值没有被检测出来，这4个被认为是missing了；而在所有的100个辛普森人物里面，只有1个被检测为异常值，被认为是False alarm，剩下的99个都认为是正常的。

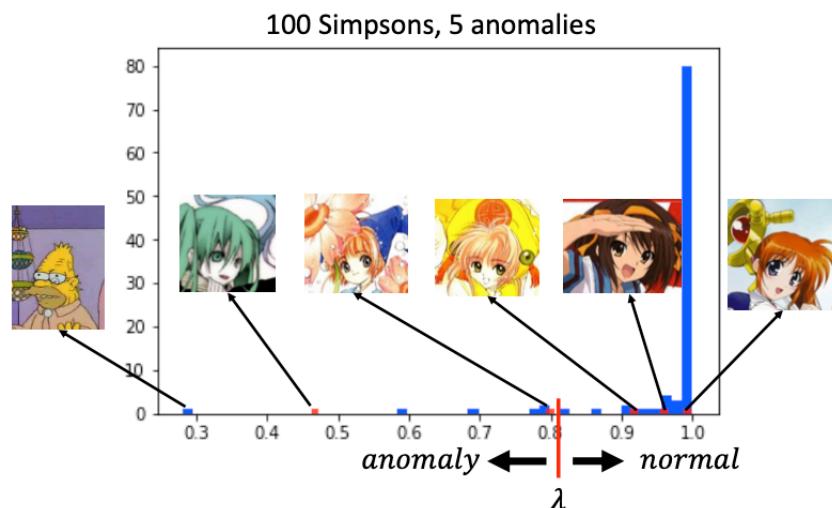
	Anomaly	Normal
Detected	1	1
Not Det	4	99

missing 100 Simpsons, 5 anomalies



如果我们现在把threshold设置为大于0.8的值，这时只有2个异常值被检测出来，还有剩下3个异常值没有被检测出来（missing）；而在所有的100个辛普森人物里面，只有6个被检测为异常值（false alarm），剩下的94个都认为是正常的。

	Anomaly	Normal		Anomaly	Normal
Detected	1	1	Detected	2	6
Not Det	4	99	Not Det	3	94



那么这两个threshold分别为0.5和0.8的异常检测系统，到底哪一个更好呢？这取决于你认为missing更严重，还是false alarm更严重。

在下图中的Cost Table A，如果是missing，就扣一分；如果是false alarm，就扣100分；如果用这种方式来衡量系统的好坏，那么左边的被扣了104分，右边的被扣了603分。

Cost Table B，如果是missing，就扣100分；如果是false alarm，就扣1分；如果用这种方式来衡量系统的好坏，那么左边的被扣了401分，右边的被扣了306分。

	Anomaly	Normal		Anomaly	Normal
Detected	1	1	Detected	2	6
Not Det	4	99	Not Det	3	94

Cost = 104

Cost = 401

Cost = 603

Cost = 306

Cost	Anomaly	Normal	Cost	Anomaly	Normal
Detected	0	100	Detected	0	1
Not Det	1	0	Not Det	100	0

Cost Table A **Cost Table B**

Some evaluation metrics consider the ranking

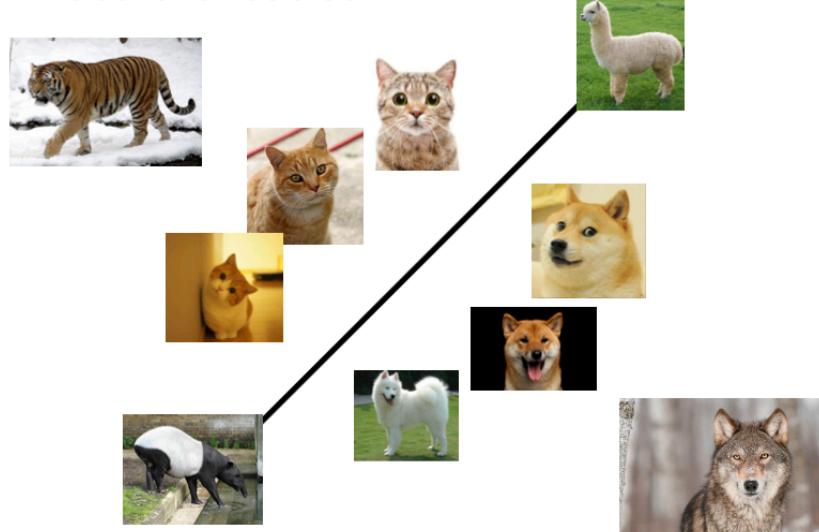
For example, Area under ROC curve

在不同的场景下，就会有不同的cost table。比如癌症检测，missing比false alarm造成的影响大得多，如果这个人患了癌症却没被检测出来，后果会非常严重，因此这种情况我们table B比较好。

Possible Issues

下图中展示了猫狗分类的例子，黑色线条表示decision boundary，靠近boundary得到的confidence score不高，如果在boundary上，就说明分类器不能确定图片的类别是猫还是狗。对于一些特别像猫的动物，比如老虎，也能够得到很好的分数，因为老虎有着和猫很像的特征，而这个特征是分类器进行分类的关键，就可以迷惑分类器，得出错误的分类结果。

Possible Issues



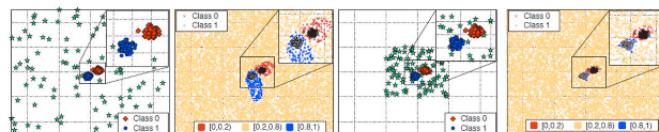
如果这个分类器判断是不是辛普森家庭人物的关键是，人物的脸和皮肤是不是黄色。那么我们可以把其他动漫人物的脸涂成黄色，就可以发现分类器判断为辛普森的概率提高了很多。



To Learn More

也有一些文献提出了解决方法，

- Learn a classifier giving low confidence score to anomaly



Kimin Lee, Honglak Lee, Kibok Lee, Jinwoo Shin, Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples, ICLR 2018

- How can you obtain anomaly?

Generating by Generative Models?

Mark Kliger, Shachar Fleishman, Novelty Detection with GAN, arXiv, 2018

Case 2: Without Labels

Twitch Plays Pokémon

先介绍一个游戏，Twitch Plays Pokémon，每个用户都可以在switch上输入自己的指令（up、left、...），可以同时有8万个用户，在这些用户中有些是异常用户，会输入无关指令干扰游戏的正常执行。



Twitch Plays Pokémon



- Why is the game so difficult?
- Probably because of “Troll” (網路小白)
 - Players that are not familiar with the game
 - Just for fun ...
 - Malicious players ...
- Assuming that most players want to complete the game (normal data for training)
 - Using anomaly detection, can we identify the “Troll” (anomaly)?
- Do we need to remove trolls?

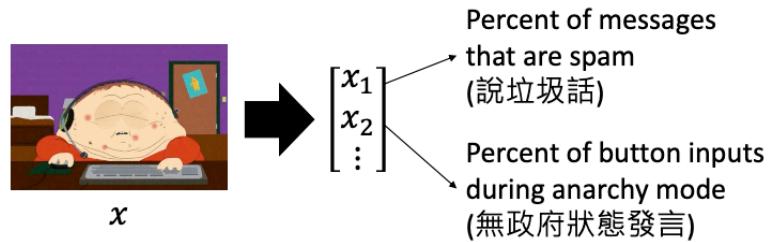
In all, the players who don't want to complete the game.

Problem Formulation

那么我们如何移除这些游戏干扰者呢？

我们首先对每一个玩家都用 x 表示，即表示成一个向量；对于其中一个维度 x_2 ，如果这个人经常发表无政府言论，我们就可以认为这个人是来游戏中捣乱的。

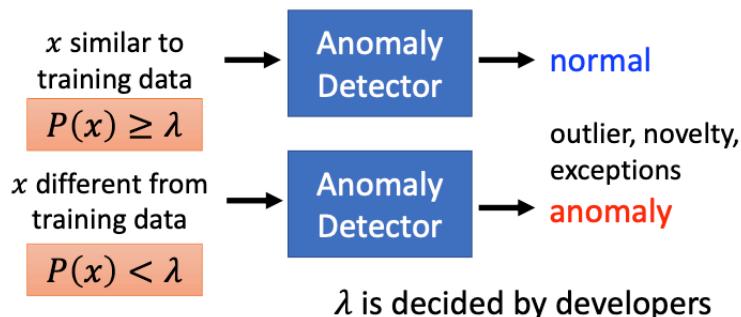
- Given a set of training data $\{x^1, x^2, \dots, x^N\}$
- We want to find a function detecting input x is *similar* to training data or not.



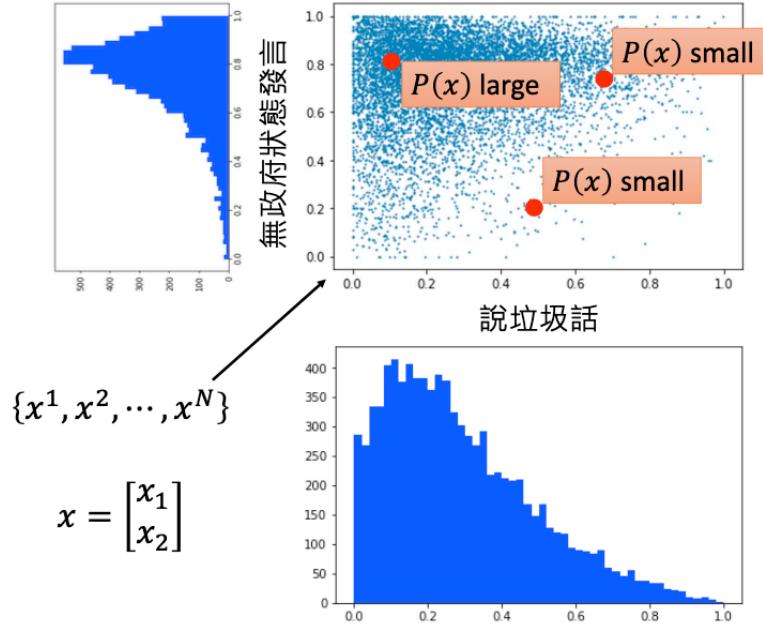
<https://github.com/ahaque/twitch-troll-detection> (Albert Haque)

但现在数据集是没有label的，没有classifier了，那么我们可以建立一个模型 $P(x)$ ，这个模型可以告诉我们发生某种行为的概率有多大。如果 $P(x) \geq \lambda$ ，那么我们就可以认为 x 是正常的，反之则认为是anomaly。

- Generated from $P(x)$**
- Given a set of training data $\{x^1, x^2, \dots, x^N\}$
 - We want to find a function detecting input x is *similar* to training data or not.



如果现在对于一个玩家用二维向量来表示，一个维度表示“说垃圾话”的几率，另一个表示“无政府状态发言”的几率。对N个玩家的数据进行可视化，可以得到下图中的散点图。如果我们对其中“说垃圾话”维度的数据进行可视化，可以发现并不是所有的玩家都不说垃圾话，大部分都会说一点点；对于第二个维度，到底有多少人的发言是在无政府状态下进行的，可以发现多数人都是在无政府状态下发言的。

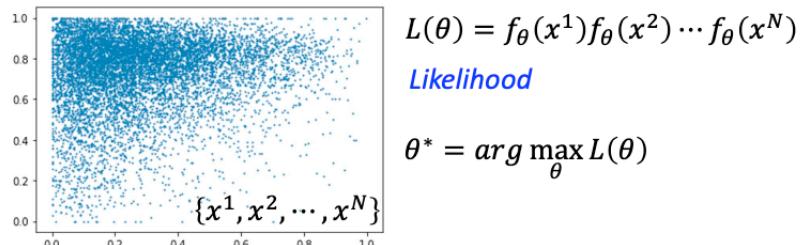


对于 $P(x)$ 比较大的玩家，我们就可以认为是normal的；对于 $P(x)$ 小的玩家，我们就认为其是anomaly的。但这并不具体，我们应该用数值化的方法来计算每一个玩家的score。

Maximum Likelihood

首先我们假设这些数据点都是从概率密度函数 $f_\theta(x)$ 中取出来的， θ 决定了概率密度函数的形状，需要从训练数据中学习出来。现在我们已经有了数据点，但不知道其概率分布 $f_\theta(x)$ 的形式。

- Assuming the data points is sampled from a probability density function $f_\theta(x)$
 - θ determines the shape of $f_\theta(x)$
 - θ is unknown, to be found from data



这里引入了likelihood的概念，表示根据我现在有的概率密度函数 $f_\theta(x^1), f_\theta(x^2), \dots, f_\theta(x^N)$ ，产生左图数据分布的概率有多大。

Q：那么我们怎么算图中的数据被产生出来的几率大小呢？

A：对于图中的所有数据点 $\{x^1, x^2, \dots, x^N\}$ ，产生数据点 x^1 的概率是 $f_\theta(x^1)$ ，产生数据点 x^2 的概率是 $f_\theta(x^2)$ ，...，产生数据点 x^N 的概率是 $f_\theta(x^N)$ ；那么对于产生所有数据点的概率，就可以用以下的式子表示，

$$L(\theta) = f_\theta(x^1)f_\theta(x^2) \cdots f_\theta(x^N)$$

也就是likelihood，和 θ 是有关的，选择不同的 θ ，就会有不同的概率密度函数，也就会有不同的likelihood，那么现在我们的任务就是找到 θ^* ，使得likelihood可以取得最大值，即

$$\theta^* = \arg \max_{\theta} L(\theta)$$

Gaussian Distribution

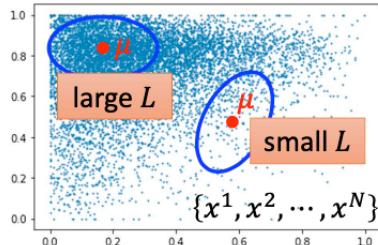
下图中的 $f_{\mu, \Sigma}(x)$ 为高斯分布，输入为 x ，输出为 x 被sample的概率，是一个数值， μ 表示均值， Σ 表示协方差矩阵。

Gaussian Distribution

D is the dimension of x

$$f_{\mu, \Sigma}(x) = \frac{1}{(2\pi)^D/2} \frac{1}{|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}$$

Input: vector x , output: probability density of sampling x
 θ which determines the shape of the function are **mean μ** and **covariance matrix Σ**



$$\begin{aligned} L(\theta) &= f_\theta(x^1) f_\theta(x^2) \cdots f_\theta(x^N) \\ L(\mu, \Sigma) &= f_{\mu, \Sigma}(x^1) f_{\mu, \Sigma}(x^2) \cdots f_{\mu, \Sigma}(x^N) \\ \theta^* &= \arg \max_{\theta} L(\theta) \\ \rightarrow \mu^*, \Sigma^* &= \arg \max_{\mu, \Sigma} L(\mu, \Sigma) \end{aligned}$$

How about $f_\theta(x)$ is from a network, and θ is network parameters? (out of the scope)

由于此时的概率密度函数发生了变化，我们现在将likelihood的形式进行变化，即

$$L(\mu, \Sigma) = f_{\mu, \Sigma}(x^1) f_{\mu, \Sigma}(x^2) \cdots f_{\mu, \Sigma}(x^N)$$

那么此时我们就需要找到 μ^*, Σ^* ，来使likelihood最大化，

$$\mu^*, \Sigma^* = \arg \max_{\mu, \Sigma} L(\mu, \Sigma)$$

在上图中，由于高斯分布的特性，数据点在 μ 附近的地方很容易被sample到，越远被sample到的几率就越低；如果 μ 在数据点很密集的地方，我们就可以认为likelihood的值很大；如果 μ 在偏离高密度的地方，likelihood会小很多。

μ^* 就是输入数据点求均值，

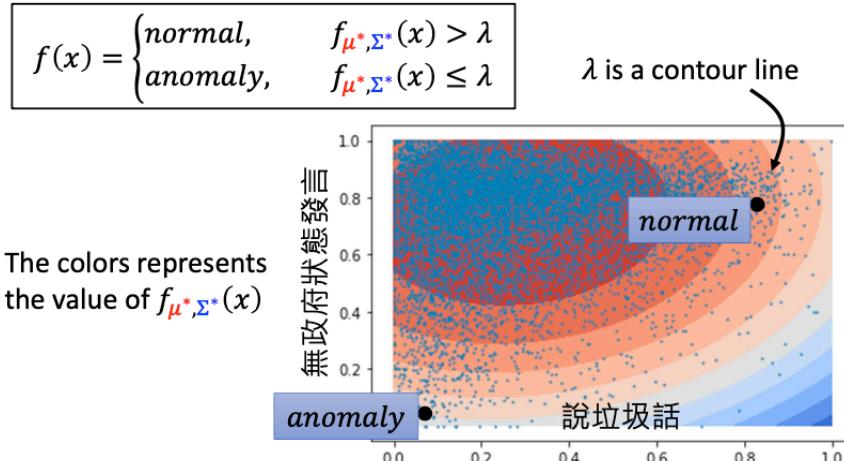
$$\begin{aligned} \mu^* &= \frac{1}{N} \sum_{n=1}^N x^n = \begin{bmatrix} 0.29 \\ 0.73 \end{bmatrix} & \Sigma^* &= \frac{1}{N} \sum_{n=1}^N (x - \mu^*)(x - \mu^*)^T \\ & & &= \begin{bmatrix} 0.04 & 0 \\ 0 & 0.03 \end{bmatrix} \end{aligned}$$

找出了对应的 μ^*, Σ^* 之后，我们就可以代入对应的概率密度函数，进行异常检测。如果 $f_{\mu^*, \Sigma^*}(x) > \lambda$ ，就认为是正常数据；如果 $f_{\mu^*, \Sigma^*}(x) \leq \lambda$ ，则认为是异常数据 (anomaly)。

如果把这个二维平面上所有的数据都输入这个概率分布，我们就可以得出下图；颜色越深，表示这越是一个一般的玩家，颜色越浅，表示异常行为越显著。

$$f_{\mu^*, \Sigma^*}(x) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma^*|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu^*)^T \Sigma^{*-1} (x - \mu^*) \right\}$$

$$\mu^* = \begin{bmatrix} 0.29 \\ 0.73 \end{bmatrix} \quad \Sigma^* = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.03 \end{bmatrix}$$



如果现在有三个玩家 $x = (x_1, x_2, x_3, x_4, x_5)^T$, 表现出了不同的行为, 根据模型算出来不同的结果。

$$f(x) = \begin{cases} \text{normal}, f_{\mu^*, \Sigma^*}(x) > \lambda \\ \text{anomaly}, f_{\mu^*, \Sigma^*}(x) \leq \lambda \end{cases}$$

More Features

- x_1 : Percent of messages that are spam (說垃圾話)
- x_2 : Percent of button inputs during anarchy mode (無政府狀態發言)
- x_3 : Percent of button inputs that are START (按 START 鍵)
- x_4 : Percent of button inputs that are in the top 1 group (跟大家一樣)
- x_5 : Percent of button inputs that are in the bottom 1 group (唱反調)

$$\begin{bmatrix} 0.1 \\ 0.9 \\ 0.1 \\ 1.0 \\ 0.0 \end{bmatrix} \rightarrow \log f_{\mu^*, \Sigma^*}(x) \rightarrow -16$$

$$\begin{bmatrix} 0.1 \\ 0.9 \\ 0.1 \\ 0.0 \\ 0.3 \end{bmatrix} \rightarrow \log f_{\mu^*, \Sigma^*}(x) \rightarrow -22$$

$$\begin{bmatrix} 0.1 \\ 0.9 \\ 0.1 \\ 0.7 \\ 0.0 \end{bmatrix} \rightarrow \log f_{\mu^*, \Sigma^*}(x) \rightarrow -2$$