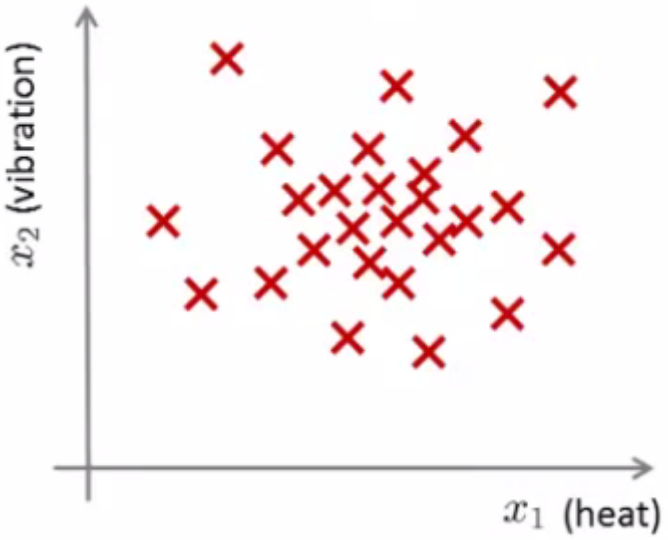
Anomaly Detection - Problem Motivation ¶

Anomaly Detection is a reasonably commonly used type machine learning, anomaly detection can be thought of as a solution to an unsupervised learning problem but has aspects of supervised learning.

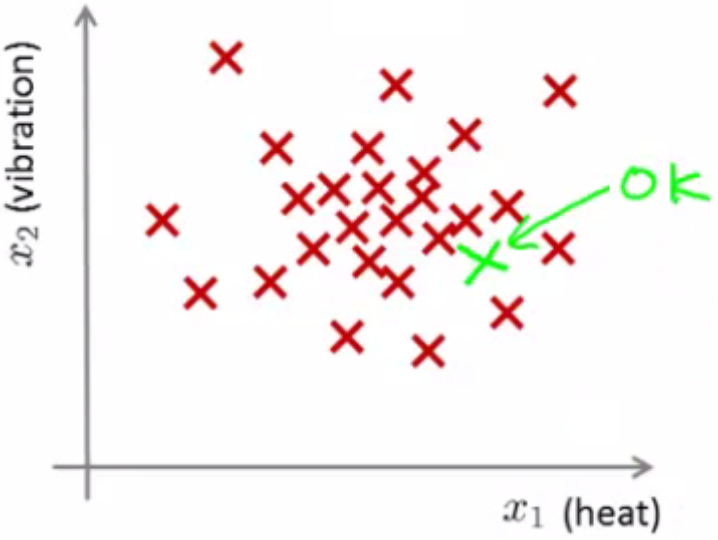
Anomaly Detection example

- Imagine we're an aircraft engine manufacturer
- As engines roll off our assembly line we're doing QA or quality assurance testing.
- Aircraft engine features:
  - x_1 = heat generated
  - x_2 = vibration intensity
  - $\dots$
- So we now have a data set of x_1 through x_m, if we have manufactured m aircraft engines (i.e. m engines were tested).
  - Dataset: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
- Say we plot that dataset:



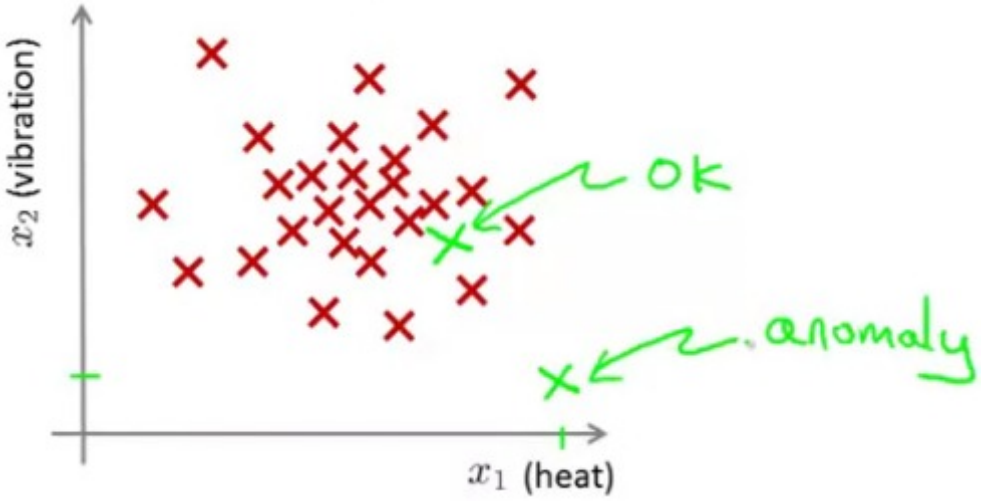
Let's say we have a New aircraft engine that rolls off the assembly line and our new aircraft engine has some set of features x_{test}

- An anomaly detection method is used to see if the new engine is anomalous (when compared to the previous engines)
- If the new engine looks like this:



That looks a lot like the aircraft engines we've seen before, and so maybe we'll say that it looks okay.

But if the engine looks like this:



If x_{test} were all the way out there, then we would call that an anomaly and maybe send that aircraft engine for further testing before we ship it to a customer, since **it looks very different** than the rest of the aircraft engines we've seen before.

Density estimation

More formally in the anomaly detection problem, we're give some data sets:

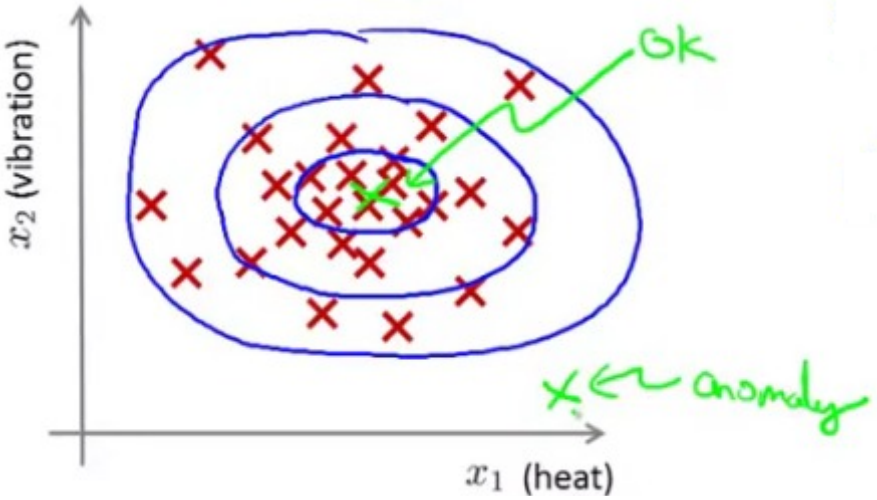
- Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, is x_{test} anomalous?

and we usually assume that these m examples are normal or non-anomalous examples, and we want an algorithm to tell us if some new example x_{test} is anomalous. Using that dataset as a reference point we can see if other examples are **anomalous**. How do we do this?

The approach that we're going to take is that given this training set, given the unlabeled training set, we're going to build a model

- Model $p(x)$: In other words, we're going to build a model for the probability of x, where x are these features of say aircraft engines.
- This asks, "What is the probability that example x is normal"
- Having built a model:
  - if $p(x_{\text{test}}) < \epsilon \rightarrow$ flag anomaly
  - if $p(x_{\text{test}}) \geq \epsilon \rightarrow$ this is OK
  - ϵ is some threshold probability value which we define, depending on how sure we need/want to be

We expect our model to (graphically) look something like this:



Applications (Anomaly Detection examples)

Fraud detection:

- $x^{(i)}$ = features of user i's activities
  - Using this data we can build a model of what normal users' activity is like
- What is the probability of "normal" behavior?
  - Identify unusual users by sending their data through the model
  - Users have activity associated with them, such as:
    - x_1 = Length on time on-line, x_2 = Location of login, x_3 = Spending frequency, …, x_4 = the typing speed of the user
- Identify unusual users by checking which have $p(x) < \epsilon$
- Flag up anything that looks a bit weird and automatically block cards/transactions

Manufactoring:

Monitoring computers in a data center.

- $x^{(i)}$ = features of machine i
- x_1 = memory use, x_2 = number of disk accesses/sec,
- x_3 = CPU load, x_4 = CPU load/network traffic.
- …

Then given the dataset of how our computers in our data center usually behave, we can model the probability of x ($p(x)$), so we can model the probability of these machines having different amounts of memory use or probability of these machines having different numbers of disk accesses or different CPU loads and so on.

If we ever have a machine whose probability of x is very small then we know that machine is behaving unusually and maybe that machine is about to go down, and we can flag that for review by a system administrator.

Video Question: Your anomaly detection system flags x as anomalous whenever $p(x) \leq \epsilon$. Suppose your system is flagging too many things as anomalous that are not actually so (similar to supervised learning, these mistakes are called false positives). What should you do?

- Try increasing ϵ.

> Try decreasing ϵ.