

앤드류 응의 머신러닝(15-3): 이상 탐지 알고리즘

σ^2

온라인 강의 플랫폼 코세라의 창립자인 앤드류 응 (Andrew Ng) 교수는 인공지능 업계의 거장입니다. 그가 스탠퍼드 대학에서 머신 러닝 입문자에게 한 강의를 그대로 코세라 온라인 강의 (Coursera.org)에서 무료로 배울 수 있습니다. 이 강의는 머신러닝 입문자들의 필수코스입니다. 인공지능과 머신러닝을 혼자 공부하면서 자연스럽게 만나게 되는 강의입니다.

Anomaly Detection

(이상 탐지)

Density Estimation (밀도 추정)

Anomaly Detection Algorithm (이상 탐지 알고리즘)

In the last video, we talked about the Gaussian distribution. In this video lets apply that to develop an anomaly detection algorithm.

지난 강의에서 배운 가우시안 분포를 배웠습니다. 이번 강의는 가우시안 분포를 적용한 이상 탐지 알고리즘을 개발합니다.

Density estimation

→ Training set: $\{x^{(1)}, \dots, x^{(m)}\}$

Each example is $x \in \mathbb{R}^n$

$p(x)$

$$= p(x_1; \mu_1, \sigma_1^2) p(x_2; \mu_2, \sigma_2^2) p(x_3; \mu_3, \sigma_3^2) \dots p(x_n; \mu_n, \sigma_n^2)$$

$$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$

$$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

$$x_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$$

Let's say that we have an unlabeled training set of M examples, and each of these examples is going to be a feature in \mathbb{R}^n so your training set could be, feature vectors from the last M aircraft engines being manufactured. Or it could be features from m users or something else. The way we are going to address anomaly detection, is we are going to model p of x from the data sets. We're going to try to figure out what are high probability features, what are lower probability types of features. So, x is a vector and what we are going to do is model p of x , as probability of x_1 , that is of the first component of x , times the probability of x_2 , that is the probability of the second feature, times the probability of the third feature, and so on up to the probability of the final feature of x_n . Now I'm leaving space here cause I'll fill in something in a minute. So, how do we model each of these terms, p of x_1 , p of x_2 , and so on. What we're going to do, is assume that the feature, x_1 , is distributed according to a Gaussian distribution, with some mean, which you want to write as μ_1 and some variance, which I'm going to write as σ_1^2 , and so p of x_1 is going to be a Gaussian probability distribution, with mean μ_1 and variance σ_1^2 . And similarly I'm going to assume that x_2 is distributed, Gaussian, that's what this little \sim stands for, that means distributed Gaussian with mean μ_2 and σ_2^2 , so it's distributed according to a different Gaussian, which has a different set of parameters, μ_2 σ_2^2 . And similarly, you know, x_3 is yet another Gaussian, so this can have a different mean and a different standard deviation than the other features, and so on, up to x_n . And so

that's my model.

레이블이 없는 m 개의 학습 셋이 있고, 학습 예제 x 는 R^n 차원입니다. n 은 피처의 개수입니다. 학습 셋은 제조 중인 m 개의 항공기 엔진의 피처 벡터이거나 m 명의 사용자의 피처 벡터일 수 있습니다. 이상 탐지는 데이터셋에서 확률 $p(x)$ 를 모델링하는 것입니다. 확률이 높은 피처와 확률이 낮은 피처를 찾아내야 합니다. x 는 피처 벡터이고 피처 별 확률 $p(x_n)$ 다음과 같습니다.

x_1 의 확률 $p(x_1)$,

x_2 의 확률 $p(x_2)$,

x_3 의 확률 $p(x_3)$,

...

x_n 의 확률 $p(x_n)$

여기에서 공간을 남겨두었습니다. 잠시 후에 채울 것입니다. 이제 $p(x_1)$, $p(x_2)$ 등의 각 피처 별 확률을 어떻게 모델링할까요? 피처 x 는 가우스 분포를 따른다고 가정합니다.

데이터셋 : $x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(m)}$

가우시안 확률 밀도 함수 : $x \sim N(x|\mu, \sigma^2)$

$$x_1 \sim N(x_1|\mu_1, \sigma_1^2)$$

$$x_2 \sim N(x_2|\mu_2, \sigma_2^2)$$

$$x_3 \sim N(x_3|\mu_3, \sigma_3^2)$$

...

$$x_n \sim N(x_n|\mu_n, \sigma_n^2)$$

이것이 데이터 셋에서 확률 $p(x)$ 를 모델링하는 것입니다. 여기 방정식은 x_1 에서 x_n 까지 피처 별로 서로 다른 평균과 표준 편차를 가집니다.

Just as a side comment for those of you that are experts in statistics, it turns out that this equation that I just wrote out actually corresponds to an

independence assumption on the values of the features x_1 through x_n . But in practice it turns out that the algorithm of this fragment, it works just fine, whether or not these features are anywhere close to independent and even if independence assumption doesn't hold true this algorithm works just fine. But in case you don't know those terms I just used independence assumptions and so on, don't worry about it. You'll be able to understand it and implement this algorithm just fine and that comment was really meant only for the experts in statistics.

이 방정식은 x_1 에서부터 x_n 까지 값은 독립 가정을 전제로 합니다. 즉, 각 값들이 상호 영향을 주지 않습니다. 실제로 피처가 독립적이던 의존적이던 상관없이 독립 가정이 사실이 아니더라도 알고리즘은 잘 작동합니다. 하지만, 독립 가정의 의미를 정확히 모르더라도 걱정하지 마세요. 여러분들은 방정식을 이해하고 알고리즘을 잘 구현할 수 있습니다. 여기서의 독립 가정을 언급한 것은 통계 전문가를 위한 것입니다.

→ Density estimation

→ Training set: $\{x^{(1)}, \dots, x^{(m)}\}$

Each example is $x \in \mathbb{R}^n$

→ $p(x)$

$$= p(x_1; \mu_1, \sigma_1^2) p(x_2; \mu_2, \sigma_2^2) p(x_3; \mu_3, \sigma_3^2) \dots p(x_n; \mu_n, \sigma_n^2) \leftarrow$$

$$= \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2)$$

$$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$

$$x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

$$x_3 \sim \mathcal{N}(\mu_3, \sigma_3^2)$$

$$\sum_{i=1}^n i = 1+2+3+\dots+n$$

$$\prod_{i=1}^n i = 1 \times 2 \times 3 \times \dots \times n$$

Finally, in order to wrap this up, let me take this expression and write it a little bit more compactly. So, we're going to write this is a product from J equals one through N , of P of X_J parameterized by μ_j comma σ_j^2 . So this funny symbol here, there is capital Greek alphabet π , that funny symbol there corresponds to taking the product of a set of values. And so, you're familiar with the summation notation, so the sum

from i equals one through n , of i . This means $1 + 2 + 3$ plus dot dot dot, up to n . Whereas this funny symbol here, this product symbol, right product from i equals 1 through n of i . Then this means that, it's just like summation except that we're now multiplying. This becomes 1 times 2 times 3 times up to N . And so using this product notation, this product from j equals 1 through n of this expression. It's just more compact, it's just shorter way for writing out this product of all of these terms up there. Since we're taking these p of x_j given μ_j comma σ_j^2 terms and multiplying them together.

마지막으로 방정식을 간결하게 작성합니다.

$$\begin{aligned} p(x) &= p(x_1 | \mu_1, \sigma_1^2) p(x_2 | \mu_2, \sigma_2^2) p(x_3 | \mu_3, \sigma_3^2) \dots p(x_n | \mu_n, \sigma_n^2) \\ &= \prod_{j=1}^n p(x_j | \mu_j, \sigma_j^2) \end{aligned}$$

여기서, 재미있는 그리스 알파벳 대문자 파이(Π)가 있습니다. 여러분이 익숙한 합산 기호 시그마(Σ)와 다릅니다. Π (파이)는 $j = 1$ 에서 n 까지 곱을 의미합니다.

$$\sum_{i=1}^n i = 1 + 2 + 3 + \dots + n$$

$$\prod_{i=1}^n i = 1 \times 2 \times 3 \times \dots \times n$$

And, by the way the problem of estimating this distribution p of x , they're sometimes called the problem of density estimation. Hence the title of the slide. So putting everything together, here is our anomaly detection algorithm. The first step is to choose features, or come up with features x_i that we think might be indicative of anomalous examples. So what I mean by that, is, try to come up with features, so that when there's an unusual user in your system that may be doing fraudulent things, or

when the aircraft engine examples, you know there's something funny, something strange about one of the aircraft engines. Choose features X_i , that you think might take on unusually large values, or unusually small values, for what an anomalous example might look like. But more generally, just try to choose features that describe general properties of the things that you're collecting data on.

Next, given a training set, of M , unlabeled examples, X_1 through X_M , we then fit the parameters, μ_1 through μ_n , and σ^2_1 through σ^2_n , and so these were the formulas similar to the formulas we have in the previous video, that we're going to use the estimate each of these parameters, and just to give some interpretation, μ_j , that's my average value of the j feature. μ_j goes in this term p of x_j . which is parametrized by μ_j and σ^2_j . And so this says for the μ_j just take the mean over my training set of the values of the j feature. And, just to mention, that you do this, you compute these formulas for j equals one through n . So use these formulas to estimate μ_1 , to estimate μ_2 , and so on up to μ_n , and similarly for σ^2 ,

And it's also possible to come up with vectorized versions of these. So if you think of μ as a vector, so μ is a vector there's μ_1 , μ_2 , down to μ_n , then a vectorized version of that set of parameters can be written like so $\sum_{i=1}^n x_i$. So, this formula that I just wrote out estimates this x_i as the feature vectors that estimates μ for all the values of n simultaneously. And it's also possible to come up with a vectorized formula for estimating σ^2_j .

그런데, 확률 $p(x)$ 를 추정하는 문제를 때때로 밀도 추정의 문제라고 부릅니다. 지금까지 설명한 이상 탐지 알고리즘을 순서대로 정리합니다.

첫 번째 단계는 비정상적인 학습 예제로 표시할 수 있는 피쳐 x_i 를 선택합니다. 즉, 시스템에서 사기 행위를 할 수도 있는 비정상적인 사용자가 있을 때 또는 항공기 엔진

에 이상한 점이 있을 때 찾아낼 수 있는 피처를 찾아냅니다. 즉, 비정상적인 예가 있다면 비정상적으로 큰 값을 취하거나 비정상적으로 작은 값을 취할 수 있는 피처 x_i 를 선택합니다. 일반적으로 데이터를 수집하는 대상들의 일반 속성을 설명하는 피처를 선택합니다.

두 번째 단계는 x_1 에서 x_m 까지 레이블이 없는 예제를 가진 학습 셋에서 가우시안 파라미터 μ_1 에서 μ_n 까지와 σ_1^2 에서 σ_n^2 까지의 파라미터의 값을 최적화합니다. 가우시안 파라미터 μ 와 σ^2 를 추정하는 함수는 다음과 같습니다.

$$\prod_{j=1}^n p(x_j | \mu_j, \sigma_j^2) \text{ 일때}$$

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$$

여기서 피처의 개수 $j = 1$ 에서 n 까지입니다. 평균을 구하는 공식을 활용하여 $\mu_1, \mu_2, \dots, \mu_n$ 까지 추정할 수 있고, 표준 편차의 제곱을 구하는 공식을 활용하여 σ^2 를 추정할 수 있습니다.

μ 와 σ^2 에 대한 추정은 벡터화 구현이 가능합니다. 벡터 $\mu = [\mu_1; \mu_2; \mu_3; \dots; \mu_n]$ 입니다. 방금 작성한 공식은 모든 값에 대해 동시에 μ 를 추정하는 피처 벡터로 추정합니다. σ_j^2 를 추정하기 위한 벡터화 구현도 가능합니다.

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

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if $p(x) < \epsilon$

Finally, when you're given a new example, so when you have a new aircraft engine and you want to know is this aircraft engine anomalous. What we need to do is then compute p of x , what's the probability of this

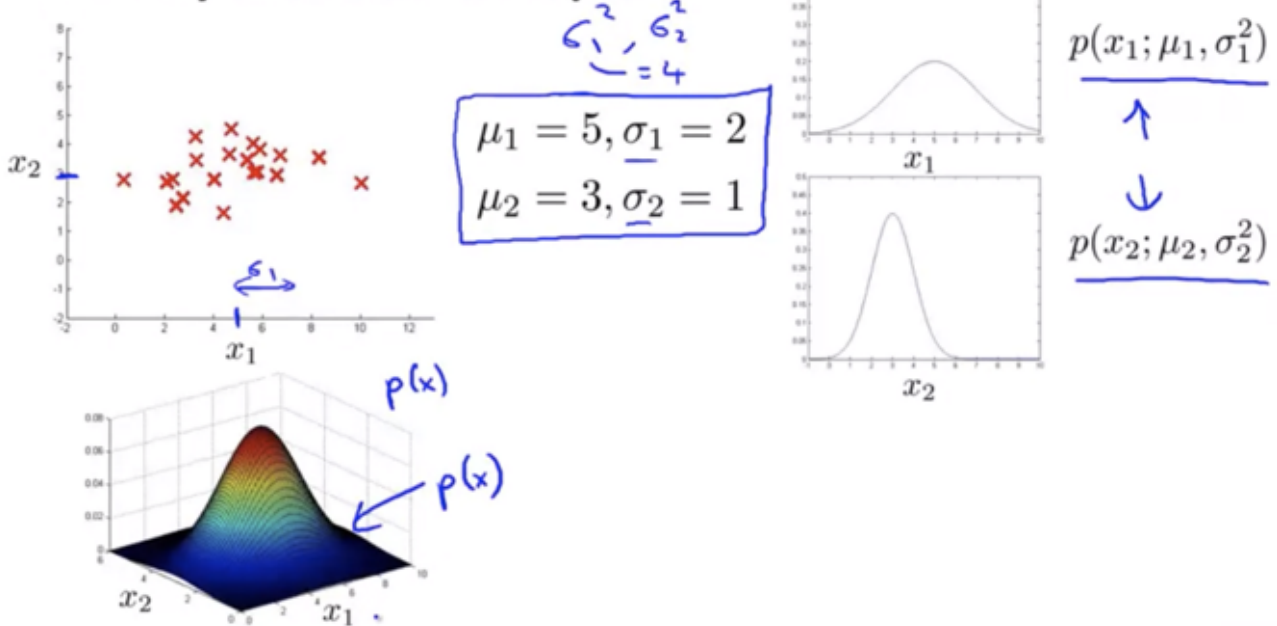
new example? So, p of x is equal to this product, and what you implement, what you compute, is this formula and where over here, this thing here this is just the formula for the Gaussian probability, so you compute this thing, and finally if this probability is very small, then you flag this thing as an anomaly.

마지막으로, 새로운 항공기 엔진의 테스트 결과와 같은 새로운 예제가 이상인지 아닌지를 판단합니다. **확률 $p(x)$ 를 계산합니다.** 새로운 예제의 확률은 얼마나 될까요? 따라서, 확률 $p(x)$ 는 다음과 같습니다.

$$\begin{aligned} p(x) &= p(x_1 | \mu_1, \sigma_1^2) p(x_2 | \mu_2, \sigma_2^2) p(x_3 | \mu_3, \sigma_3^2) \dots p(x_n | \mu_n, \sigma_n^2) \\ &= \prod_{j=1}^n p(x_j | \mu_j, \sigma_j^2) \\ &= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right) \end{aligned}$$

이것은 가우시안 확률에 대한 공식입니다. 이것을 계산하고 확률이 매우 작다면 비정상으로 표시합니다.

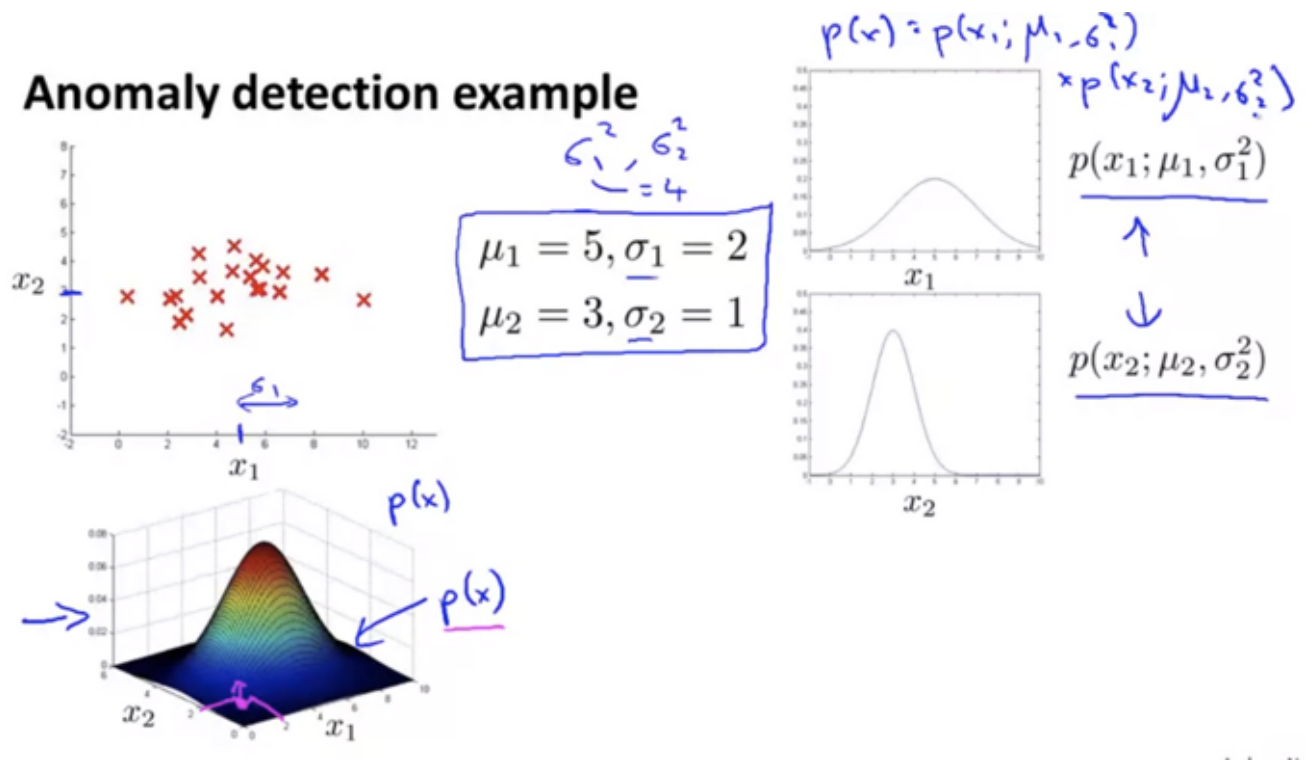
Anomaly detection example



Here's an example of an application of this method. Let's say we have this data set plotted on the upper left of this slide. If you look at this, well, let's look at the feature of x_1 . If you look at this data set, it looks like on average, the features x_1 has a mean of about 5 and the standard deviation, if you only look at just the x_1 values of this data set has the standard deviation of maybe 2. So that sigma 1 and looks like x_2 the values of the features as measured on the vertical axis, looks like it has an average value of about 3, and a standard deviation of about 1. So if you take this data set and if you estimate μ_1 , μ_2 , σ_1 , σ_2 , this is what you get. And again, I'm writing sigma here, I'm think about standard deviations, but the formula on the previous 5 actually gave the estimates of the squares of these things, so sigma squared 1 and sigma squared 2. So, just be careful whether you are using sigma 1, sigma 2, or sigma squared 1 or sigma squared 2. So, sigma squared 1 of course would be equal to 4, for example, as the square of 2.

여기 이상 탐지 알고리즘을 적용한 사례입니다. 왼쪽 상단은 데이터 셋이 그려져 있습니다. 수평축 피쳐 x_1 의 평균 μ_1 은 5이고, 표준 편차 σ_1^2 를 추정합니다. 이 데이터 셋의 피쳐 x_1 의 표준 편차 σ_1 는 2입니다. 수직축 x_2 의 평균 μ_2 는 3이고, 표준 편차 $\sigma_2 = 1$ 입니다. 파란색 박스의 값입니다. 실제로 확률 $p(x_j; \mu_j, \sigma_j^2)$ 은 표준 편차

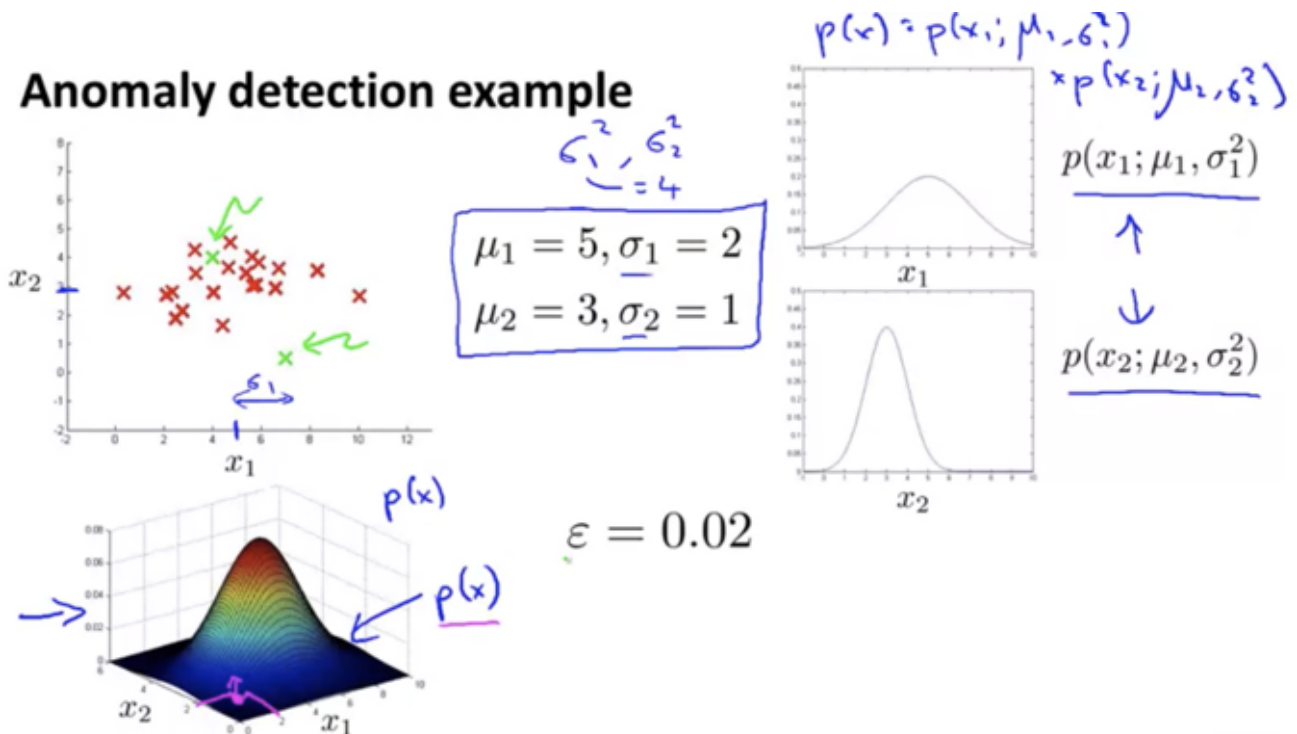
의 제곱을 사용합니다. 따라서, $\sigma_1^2 = 4$, $\sigma_2^2 = 1$ 입니다.



And in pictures what p of x_1 parametrized by μ_1 and σ_1^2 and p of x_2 , parametrized by μ_2 and σ_2^2 , that would look like these two distributions over here. And, turns out that if we were to plot p of x , right, which is the product of these two things, you can actually get a surface plot that looks like this. This is a plot of p of x , where the height above of this, where the height of this surface at a particular point, so given a particular x_1 x_2 values of x_2 if x_1 equals 2, x equal 2, that's this point. And the height of this 3-D surface here, that's p of x . So p of x , that is the height of this plot, is literally just p of x_1 parametrized by μ_1 σ_1^2 , times p of x_2 parametrized by μ_2 σ_2^2 . Now, so this is how we fit the parameters to this data.

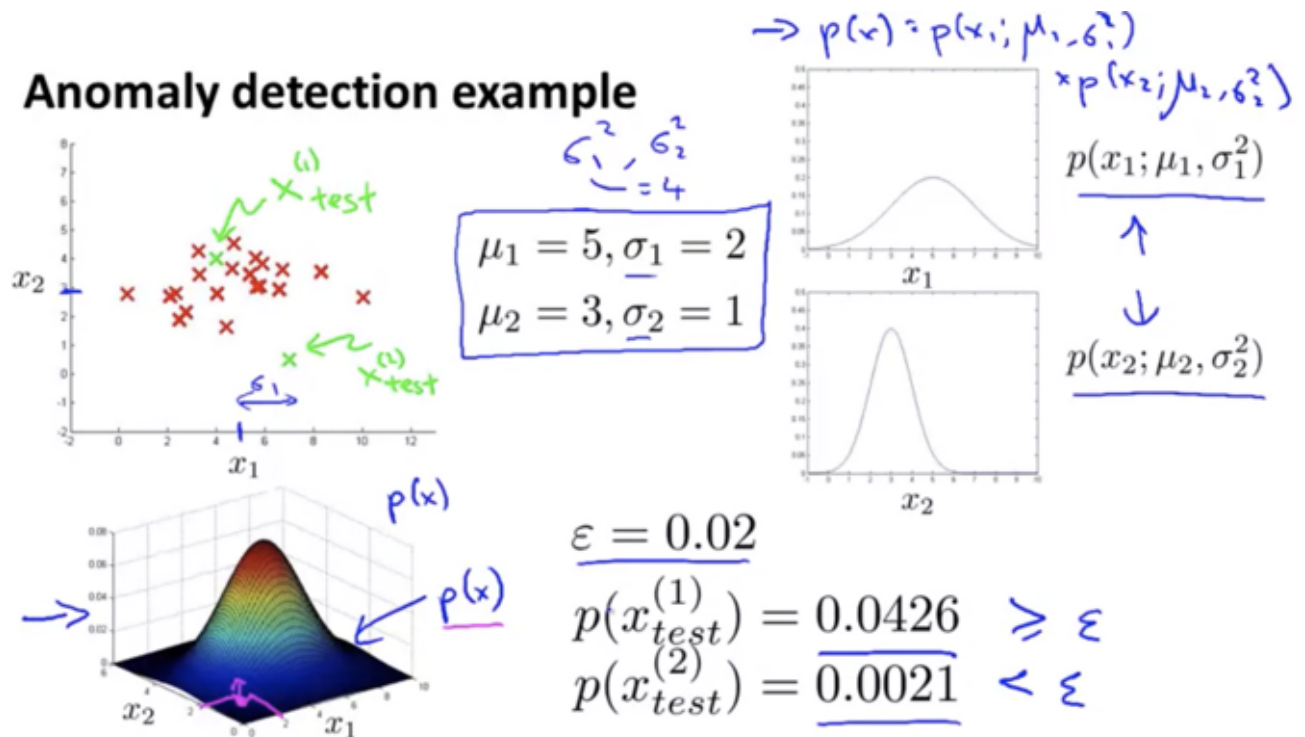
그리고 우측 상단의 첫 번째 사진은 $p(x_1; \mu_1, \sigma_1^2)$ 이고, 우측 상단의 두 번째 사진은 $p(x_2; \mu_2, \sigma_2^2)$ 입니다. 두 피쳐의 확률의 곱을 도식화하면 좌측 하단의 그래프입니다. 그래프는 $p(x)$ 입니다. 특정 지점에서 높이는 이 표면의 높이입니다. $x_1 = 2$ 이고, $x_2 = 2$ 이면 만나는 이 지점은 $x_1 x_2$ 입니다. 3차원 표면의 높이는 $p(x)$ 입니다. 즉, **3차원 그래프의 높이는 문자 그대로 다음과 같습니다.**

이것이 데이터에 파라미터를 맞추는 방법입니다.



Let's see if we have a couple of new examples. Maybe I have a new example there. Is this an anomaly or not? Or, maybe I have a different example, maybe I have a different second example over there. So, is that an anomaly or not? The way we do that is, we would set some value for Epsilon, let's say I've chosen Epsilon equals 0.02. I'll say later how we choose Epsilon.

몇 가지 새로운 예제를 살펴봅시다. 아마도 왼쪽 상단의 그림에 새로운 예제가 있을 것입니다. 중앙의 녹색 점인 첫 번째 예제는 비정상입니까? 아니면 중앙에서 멀리 떨어진 녹색 점인 두 번째 예제는 비정상입니까? 이렇게 하는 방식이 엡실론(ϵ)에 대한 값을 설정하는 것입니다. ϵ 을 0.02로 선택했다고 가정합니다. 나중에 ϵ 을 선택하는 방법에 대해 설명합니다.

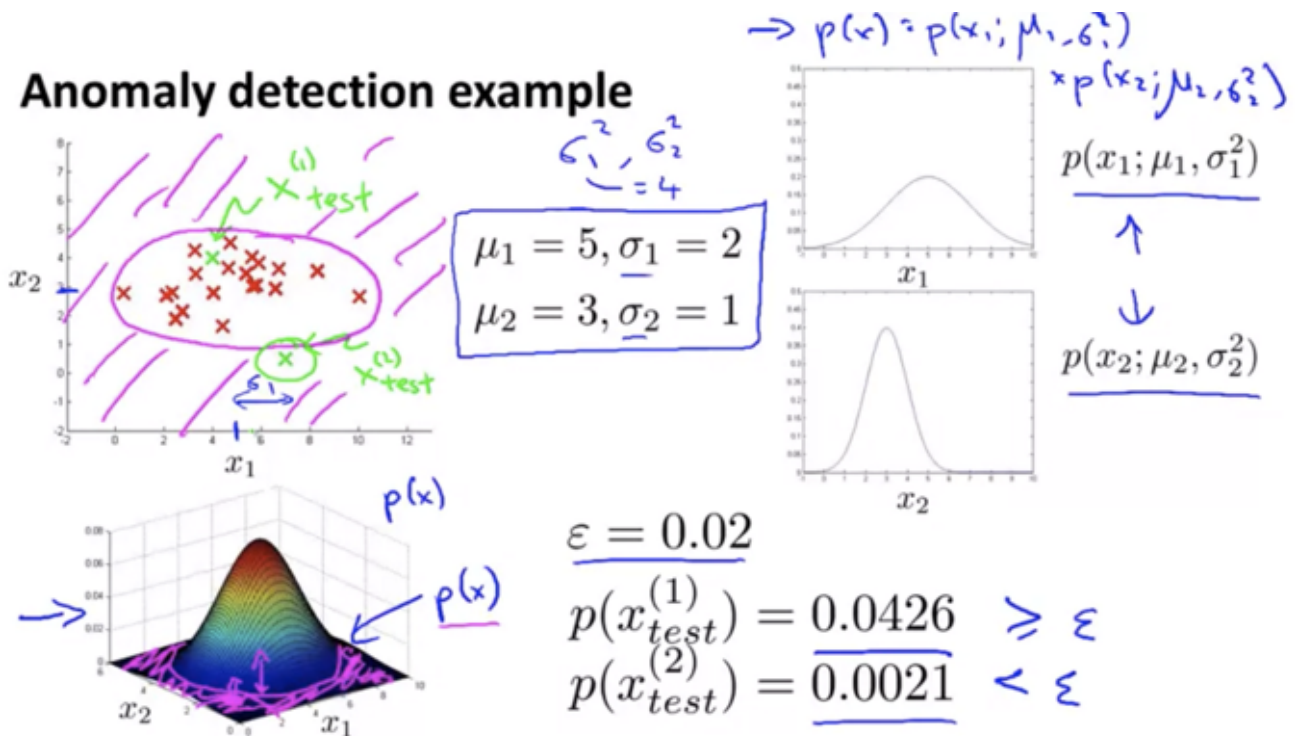


But let's take this first example, let me call this example X1 test. And let me call the second example X2 test. What we do is, we then compute p of X1 test, so we use this formula to compute it and this looks like a pretty large value. In particular, this is greater than, or greater than or equal to epsilon. And so this is a pretty high probability at least bigger than epsilon, so we'll say that X1 test is not an anomaly. Whereas, if you compute p of X2 test, well that is just a much smaller value. So this is less than epsilon and so we'll say that that is indeed an anomaly, because it is much smaller than that epsilon that we then chose. And in fact, I'd improve it here. What this is really saying is that, you look through the 3d surface plot. It's saying that all the values of x_1 and x_2 that have a high height above the surface, corresponds to an a non-anomalous example of an OK or normal example.

첫 번째 예제는 $x_{test}^{(1)}$ 이라고 하고, 두 번째 예제는 $x_{test}^{(2)}$ 라고 합시다. $x_{test}^{(1)}$ 의 확률 $p(x_{test}^{(1)})$ 를 계산합니다. $p(x_{test}^{(1)})$ 를 계산하기 위해서는 아래 공식을 사용합니다.

$p(x_{test}^{(1)}) = 0.0426$ 으로 상당히 큰 값처럼 보입니다. ε 보다 크거나 같으므로 $x_{test}^{(1)}$ 이 이상이 아니라고 판단할 수 있습니다. 반면에 $p(x_{test}^{(2)}) =$

0.0021로 상당히 작은 값입니다. ϵ 보다 작거나 같습니다. 실제로 $x_{test}^{(2)}$ 는 이상이라고 판단할 수 있습니다. 이것이 3차원 그래프 본 것입니다. 표면 위의 높이가 $\epsilon = 0.02$ 보다 높은 x_1 및 x_2 의 모든 값이 정상적인 예제입니다.



Whereas all the points far out here, all the points out here, all of those points have very low probability, so we are going to flag those points as anomalous, and so it's gonna define some region, that maybe looks like this, so that everything outside this, it flags as anomalous, whereas the things inside this ellipse I just drew, if it considers okay, or non-anomalous, not anomalous examples. And so this example x_2 test lies outside that region, and so it has very small probability, and so we consider it an anomalous example.

반면에 여기 아래에 있는 모든 점들, $\epsilon = 0.02$ 보다 낮은 모든 점들은 비정상적인 점들로 표시합니다. 아마도 이렇게 보일 것입니다. 이차원 그래프로 본다면 타원 밖에 있는 모든 것들은 비정상적인 것으로 표시되는 반면에 타원 안에 있는 모든 것들은 정상으로 표시합니다. 그래서 $x_{test}^{(2)}$ 예제는 타원 바깥에 있으므로 확률이 매우 낮습니다. 그래서 이상이 있는 예제라고 판단합니다.

In this video we talked about how to estimate p of x , the probability of x ,

for the purpose of developing an anomaly detection algorithm. And in this video, we also stepped through an entire process of giving data set, we have, fitting the parameters, doing parameter estimations. We get μ and σ parameters, and then taking new examples and deciding if the new examples are anomalous or not. In the next few videos we will delve deeper into this algorithm, and talk a bit more

about how to actually get this to work well.

이번 강의에서 이상 탐지 알고리즘을 개발하기 위해 x 의 확률인 $p(x)$ 를 추정하는 방법을 설명했습니다. 그리고, 데이터 셋에 대한 전체 프로세스를 단계별로 짚어 보았습니다. 파라미터를 맞추고 파라미터를 추정하였습니다. μ 와 σ 파라미터의 값을 계산하고 새로운 예제에 적용하여 정상인지 비정상인지를 판단합니다. 다음 몇 개의 강의에서 알고리즘에 대해 더 자세히 알아보고 실제로 작동하는 방식을 공부합니다.

앤드류 응의 머신러닝 동영상 강의

정리하며

이상 탐지 알고리즘은 어떤 행동이나 물체가 정상인지 비정상인지를 탐지합니다. 수학적으로 데이터 셋에서 확률 $p(x)$ 를 모델링합니다. 새로운 예제 $p(x_{test})$ 가 엡실론(ϵ) 보다 작으면 이상으로 표시합니다. 반면에 $p(x_{test})$ 가 엡실론(ϵ) 보다 크면 정상으로 표시합니다.

이상 탐지 알고리즘을 개발하는 순서는 다음과 같습니다.

1) 비정상적이거나 이상으로 구별할 수 있는 피쳐 x_i 를 결정합니다.

시스템에 사기 행위를 할 수도 있는 비정상적인 사용자나 비정상 항공기 엔진을 선별할 수 있는 피쳐를 정의합니다. 수학적으로 비정상적인 예는 피쳐 x_i 에 대해 비정상적으로 큰 값을 취하거나 비정상적으로 작은 값을 나타낼 것입니다.

2) $\mu_1, \mu_2, \dots, \mu_n$ 까지 파라미터와 $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$ 까지의 파라미터 값을 추정합니다.

시스템에서 두 파라미터의 값을 추정하는 방법은 다음과 같습니다.

$$\prod_{j=1}^n p(x_j | \mu_j, \sigma_j^2) \text{ 일때}$$

$$\mu = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$

$$\sigma^2 = \frac{1}{m} \sum_{j=1}^m (x^{(i)} - \mu)^2$$

3) 새로운 예제에 대해 모든 피처에 대한 확률 $p(x)$ 을 계산합니다.

새로운 예제가 주어졌을 때 즉 새로운 항공기 엔진이 주어졌을 때 이상이 있는 지를 판단합니다.

$$\begin{aligned} p(x) &= p(x_1 | \mu_1, \sigma_1^2) p(x_2 | \mu_2, \sigma_2^2) p(x_3 | \mu_3, \sigma_3^2) \dots p(x_n | \mu_n, \sigma_n^2) \\ &= \prod_{j=1}^n p(x_j | \mu_j, \sigma_j^2) \\ &= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right) \end{aligned}$$

4) 새로운 예제가 이상인지 정상인지를 판단합니다.

$p(x_{test})$ 가 엡실론(ϵ) 보다 작으면 이상으로, 엡실론(ϵ) 보다 크면 정상으로 표시합니다. 이차원 그래프로 본다면 타원 밖에 있는 점들은 이상으로 표시하고, 타원 안에 있는 점들은 것들은 정상으로 표시합니다. 삼차원 그래프로 본다면 엡실론(ϵ) 보다 낮은 높이를 가진 점들은 이상으로 표시하고, 엡실론(ϵ) 보다 높은 값을 가진 점들은 정상으로 표시합니다.

$p(x_{test}) < \varepsilon$: anomaly

$p(x_{test}) \geq \varepsilon$: OK

문제풀이

데이터 셋이 주어졌을 때, μ_j 와 σ_j^2 를 어떻게 추정할까요?

Given a training set $\{x^{(1)}, \dots, x^{(m)}\}$, how would you estimate each μ_j and σ_j^2 (Note $\mu_j \in \mathbb{R}, \sigma_j^2 \in \mathbb{R}$.)

- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m x^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$
- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)})^2, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$
- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$
- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$

정답은 4번입니다.