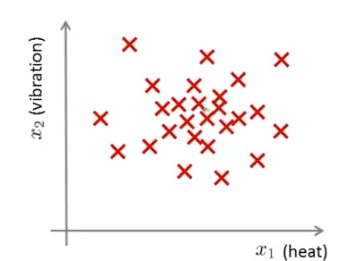
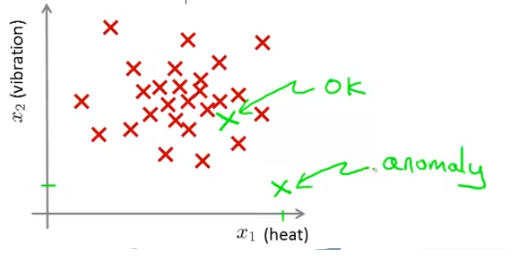
Detección de anomalías

# Motivación del problema

En la siguiente serie de vídeos, me gustaría contarles acerca de un problema que se llama detección de anomalías. Este es un uso común de aprendizaje automático. Uno de los aspectos interesantes es que es principalmente para problemas no supervisados pero también hay algunos aspectos de este problema que son muy similares a una especie de problema de aprendizaje supervisado.   
  
Así que, ¿qué es la detección de anomalías? Para explicarlo, déjeme usar este ejemplo de motivación. Imagine que usted es un fabricante de motores de aviones, y vamos a decir que a medida que sus motores de avión salen de la línea de ensamble usted está haciendo, ya sabe, un control de calidad o pruebas de control de calidad, y como parte de las pruebas, mide las variables de su motor de aeronave, como quizá, el calor generado, aspectos como las vibraciones y así sucesivamente. Comparto algunos amigos que trabajaban en este problema hace mucho tiempo y estos fueron en realidad los tipos de variables que fueron colectando sobre motores de avión reales, de modo que ahora tiene un conjunto de datos, desde x1 a xm, si usted tiene "m"motores de aviones fabricados y si dibuja sus datos, tal vez se parezca a esto:



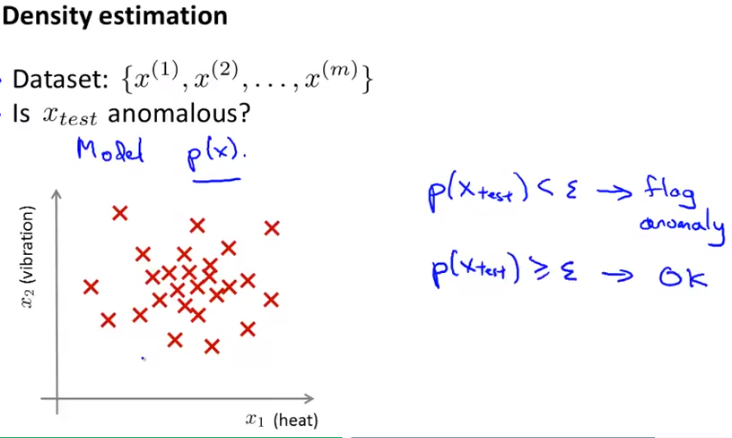
Entonces, cada punto de aquí, cada cruz mostrada es como uno de sus ejemplos valores no asignados. Así, el problema de detección de anomalías es el siguiente: Digamos que, al día siguiente, usted tendrá un nuevo motor de avión que sale de la línea de ensamble y su nuevo motor de avión tiene un conjunto de variables de "x prueba". El problema de detección de anomalías es que queremos saber si este motor de avión es anómalo de alguna manera, en otras palabras, queremos saber si, tal vez, este motor debe someterse a más pruebas o si se ve como un motor bien terminado y está listo para simplemente enviarlo a un cliente sin ninguna otra prueba.   
Por lo tanto, si su nuevo motor de avión luce como un punto dentro de los valores normales, que se parece mucho a los motores de aviones que hemos visto antes, por lo que tal vez vamos a decir que se ve bien. Considerando que, si su nuevo motor de avión, si "x prueba", fuera un punto que está lejos de los valores normales que toman las variables , entonces si x1 y x2 son las variables de este nuevo ejemplo, si las "x prueba" estuvieran todas lejos, entonces podríamos llamar a eso una anomalía



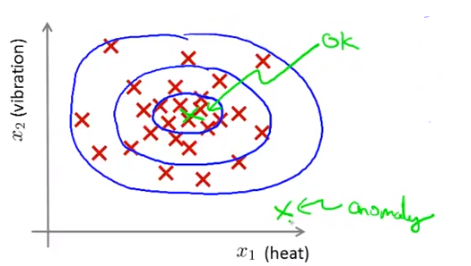
Y tal vez enviaría ese motor de avión para realizar más pruebas antes de enviarlo a un cliente, ya que se ve muy diferente al resto de los motores de aviones que hemos visto antes. Más formalmente, en el problema de detección de anomalías, vamos a dar algunos conjuntos de datos, ejemplos de x1 a xm y con frecuencia asumimos que estos ejemplos m son normales o ejemplos no anómalos, y nosotros queremos un algoritmo que nos indique si algún nuevo ejemplo"x prueba" es anómalo.

El enfoque que vamos a tomar es que, dado ub grupo de entrenamiento, dado el grupo de entrenamiento de valores no asignados, vamos a construir un modelo para "p(x)". En otras palabras, vamos a construir un modelo para la probabilidad de "x", donde x son estas variables de, por ejemplo, los motores de los aviones. Y así, después de haber construido un modelo de la probabilidad de "x", entonces vamos a decir que para el un nuevo motor de avión, si "p" de x prueba es menor que cierto épsilon, entonces marcamos esto como una anomalía.

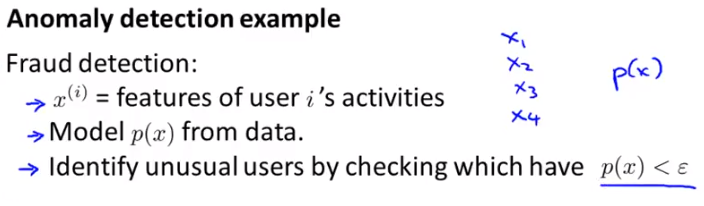
Vemos, pues, un nuevo motor que tiene muy baja probabilidad bajo un modelo de p(x), que se estima a partir de los datos, entonces nosotros señalamos esta anomalía, mientras que si p(x prueba) es mayor que o igual a un pequeño umbral, entonces decimos que está bien.



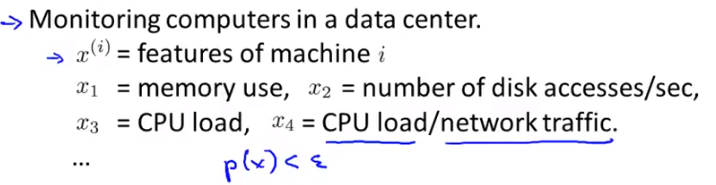
Dado el conjunto de entrenamiento, como se traza aquí, si se genera un modelo, espero que encuentre esos motores de los aviones, o espero que el modelo de p(x) indique los puntos que están ya sabe, en algún lugar en medio, esa es una probabilidad bastante alta, mientras que los puntos un poco más lejos tienen una menor probabilidad. Los puntos que están aún más lejos tiene una menor probabilidad y el punto que está allá, el punto cuyo camino está fuera, sería una anomalía.



mientras que el punto que está dentro, justo en el centro, esto estaría bien porque p(x) justo en medio de esto sería muy alto porque hemos visto un montón de puntos en esa región.   
  
Estos son algunos ejemplos de aplicaciones de detección de anomalías. Tal vez la aplicación más común de la detección de anomalías es en realidad para la detección, si tiene muchos usuarios, y si cada uno de sus usuarios toma diferentes actividades, usted sabe, quizá en su sitio web o en la planta física o algo, puede calcular las variables xi de las diferentes actividades de los usuarios y lo que puede hacer es construir un modelo, por ejemplo, de cuál es la probabilidad de que diferentes usuarios se comporten de diferentes maneras, cuál es la probabilidad de un vector particular de variables del comportamiento de los usuarios. Así que, usted sabe, ejemplos de variables de la actividad de los usuarios, puede ser que en el sitio web habrá cosas como, por ejemplo, x1 que es la frecuencia en que el usuario se conecta, x2, por ejemplo, tal vez sea el número de las páginas visitadas, o el número de transacciones, tal vez x3 sea, usted sabe, el número de mensajes de los usuarios en el foro, la variable x4 podría calcular cuál es la velocidad de escritura del usuario y algunos sitios web en realidad pueden hacer un seguimiento de esto, saber cual es la velocidad de escritura de este usuario en caracteres por segundo. Entonces puede modelar p(x) en base a este tipo de datos.



Por último, al tener su modelo p(x), puede tratar de identificar a los usuarios que se comportan de manera extraña en su sitio web mediante la revisión de cuáles tienen probabilidad de x menor a épsilon y tal vez enviar los perfiles de esos usuarios para su revisión posterior o pedir identificación adicional de esos usuarios, o algo así para protegerse de, usted sabe, un comportamiento extraño o conducta fraudulenta en su sitio web. Este tipo de técnica tenderá a marcar a los usuarios que se se están comportando de forma inusual, no sólo usuarios que tal vez se comporten de manera fraudulenta. Así que no sólo constantemente cuentas robadas o usuarios que están tratando de causar problemas o simplemente encontrar a los usuarios no habituales pero esta es realmente la técnica utilizada por muchos en sitios web que venden cosas para tratar de identificar a los usuarios que se comportan de forma extraña, que podría ser indicativo de, ya sea un comportamiento fraudulento o de cuentas de equipo que hayan sido robadas.   
  
Otro ejemplo de la detección de anomalías es la fabricación. Así que ya hablamos sobre los motores de avión en donde puede encontrar por ejemplo, motores de avión inusuales y enviarlos para una revisión adicional. Una tercera aplicación sería la supervisión de equipos en un centro de datos. De hecho, tengo algunos amigos que trabajan en esto también. Por lo que sí tiene muchas máquinas en un clúster de ordenadores o en un centro de datos, podemos hacer cosas como calcular las variables en cada máquina. Así que tal vez algunas variables capturando usted sabe, la cantidad de memoria utilizada, el número de accesos al disco, la carga del CPU, así como variables más complejas como cúal es la carga de CPU en esta máquina dividida entre la cantidad de tráfico de red en esta máquina. Entonces, dado el conjunto de datos de la forma en que se comportan comúnmente los equipos en su centro de datos, puede modelar la probabilidad de "x", puede modelar la probabilidad de que estas máquinas tengan diferentes cantidades de uso de la memoria o la probabilidad de estas máquinas teniendo diferentes números de accesos a disco o diferentes cargas de CPU y así sucesivamente. Si alguna vez tiene una máquina cuya probabilidad de "x", p(x), es muy pequeña, entonces sabrá que la máquina se está comportando inusualmente y tal vez esa máquina está a punto de descomponerse y usted puede marcarla para que el administrador del sistema la revise.



Esto en realidad se usa hoy en día en diversos centros de datos para rastrear las irregularidades que suceden en sus máquinas.

Así que eso es la detección de anomalías.   
En el siguiente video, voy a hablar un poco acerca de la distribución de Gauss y de las propiedades de revisión de la distribución de probabilidad gaussiana y en videos posteriores, aplicaré esto para desarrollar un algoritmo de detección de anomalías.

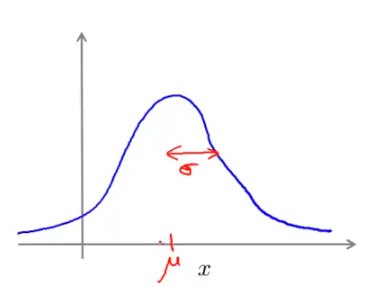
# Distribución gausiana

En este video, me gustaría hablar acerca de la distribución Gaussiana, también conocida como distribución normal. Empezaremos a aplicar la distribución Gaussiana para desarrollar un algoritmo de detección de anomalías.   
  
Digamos que "x" es un variable aleatoria de valor real, "x" es un número real. Si la distribución de probabilidad de "x" es Gaussiana, esto significa que:



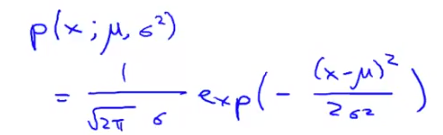


La tilde que significa "distribuido como" y para denotar la distribución Gaussiana, a veces va a escribir "N", paréntesis, «Mu», «sigma» al cuadrado. Entonces, esta letra "N" significa que es normal ya que la distribución Gaussiana y normal hacen referencia a lo mismo, así que son sinónimos y una distribución Gaussiana se parametriza mediante 2 parámetros, por un parámetro que denotamos «Mu» y un parámetro de variación que denota por «sigma» cuadrado.   
  
Si graficamos la distribución Gaussiana o densidad de probabilidad Gaussiana, se verá como la curva en forma de campana, que quizá ya haya visto antes.



Esta curva en forma de campana es parametrizada por esos 2 parámetros «Mu» y «sigma» y la ubicación del centro de esta curva en forma de campana es la «Mu» media y el ancho de esta curva en forma de campana que es más o menos, digamos, este parámetro «sigma» también llamado desviación estándar. Y así, esto especifica la probabilidad de "x" tomando diferentes valores, entonces "x" asume valores, como sabe, en el centro, se trata de valores muy altos, ya que la densidad Gaussiana aquí es bastante alta, mientras que si "x" toma valores más y más alejados, va estar disminuyendo en probabilidad.

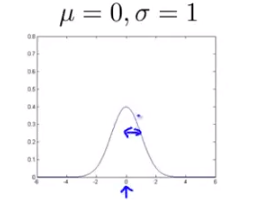
Por último, para completar, déjeme escribir la fórmula para la distribución Gaussiana, Entonces la probabilidad de x, y algunas veces escribiré esto como p (x) cuando escribimos esto como P (x; mu, sigma al cuadrado), y la formula para la distribución gaussina es:



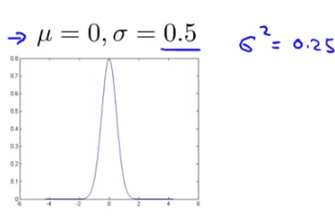
Esta formula es simplemente la curva con forma de campana. Es lo que tiene si toma un mu y un sigma fijo y dibuja P(x) así que esa campana es en realidad P(x) dibujada como una función de x para un valor fijo de “Mu” y “sigma”.

Sigma al cuadrado es lo que denominamos varianza. A veces es más facil pensar en terminos de sigma que es la llamada desviación típica y especifica el ancho de la función de densidad gaussiana. Mientras que sigma al cuadrado es conocida como la varianza. Veamos cómo se ven algunos ejemplos:

Si «Mu» es igual a cero, «sigma» es igual a 1, tenemos una distribución Gaussiana, que se centra alrededor de cero, porque eso es «Mu» y el ancho de esta Gaussiana, es la desviación de estándar, es «sigma» en aquel lado.

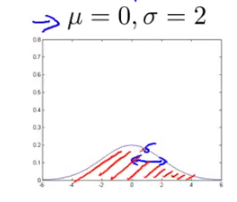


Aquí hay otro ejemplo:  
Digamos que «Mu» es igual a cero y «sigma» es igual a la mitad, y la desviación estándar es 0.5 y la varianza «sigma» al cuadrado por lo tanto sería el cuadrado de 0.5, sería entonces, 0.25. El cuadrado de 0.5 sería 0.25 y en ese caso, la distribución Gaussiana, o la densidad de probabilidad Gaussiana se ve así:



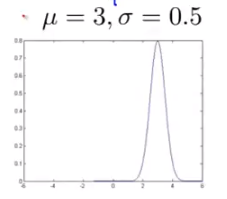
También se centra en cero pero ahora, el ancho de esto es mucho menor debido a la menor varianza, el ancho de esta densidad Gaussiana, es aproximadamente la mitad de amplio. Pero como esta es una distribución de probabilidad, el área bajo la curva, que es el área sombreada de allá, esa área (si hacemos su integral) debe ser igual a 1. Esta es una propiedad de las distribuciones de probabilidad. Porque, como sabes, esto es una densidad Gaussiana mucho más alta porque es la mitad de ancho, con la mitad de la desviación de estándar, pero es dos veces más alta.

Otro ejemplo, si «sigma» es igual a 2, obtenemos una densidad Gaussiana más robusta, o mucho más amplia. Entonces, aquí el parámetro «sigma» controla que esta densidad Gaussiana tenga un ancho mayor.



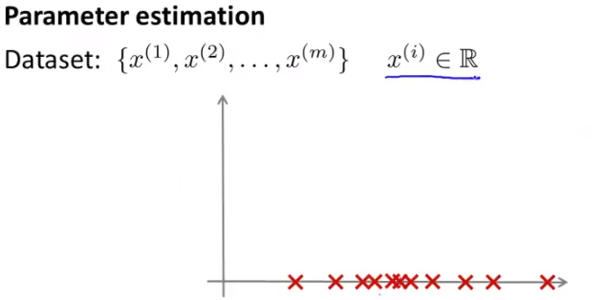
Y una vez más, el área bajo la curva, que es el área sombreada siempre se integra a 1. Esa es una propiedad de las distribuciones de probabilidad y debido a que es más amplia, también es la mitad de alta, con el fin de simplemente integrarse a la misma cosa.

Finalmente, un último ejemplo sería, si ahora cambiamos los parámetros «Mu» también, entonces en vez de centrarse en cero, ahora tenemos una distribución Gaussiana centrada en tres, porque se desplaza sobre toda la distribución Gaussiana.

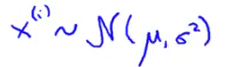


## Problema de estimación del parámetro

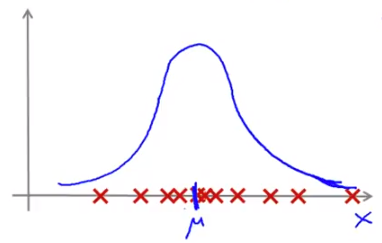
A continuación, hablemos sobre el problema de estimación de parámetros.   
  
¿Qué es el problema de estimación parámetro? Supongamos que tenemos un conjunto de datos de ejemplos "m", y digamos que cada uno de estos ejemplos es un número real.



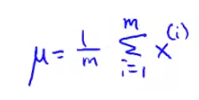
Aquí en la figura, he trazado un ejemplo de un conjunto de datos, donde el eje horizontal es el eje "x" y, como sabe, tengo un rango de ejemplos de "x" y los acabo de trazar en esta figura. Y el problema de estimación de parámetros es, digamos que tengo la sospecha de que estos ejemplos de una distribución Gaussiana, entonces tengo la sospecha de que cada uno de mis ejemplos x(i) se distribuye según una distribución normal o una distribución Gaussiana, con cierto parámetro «Mu» y algún parámetro «sigma» cuadrado:



pero no sé cuáles son los valores de estos parámetros. El problema con la estimación de parámetros es: dado mi conjunto de datos que quiero encontrar, quiero estimar, cuáles son los valores de «Mu» y «sigma» cuadrada. Así que si tiene un conjunto de datos como este, parece que tal vez, sí estimo de qué distribución Gaussiana vinieron los datos, tal vez eso puede ser más o menos la distribución Gaussiana de donde provinieron, con «Mu» siendo el centro de la distribución y «sigma», la desviación estándar que controla la anchura de esta distribución Gausiana.



Parece un ajuste razonable de los datos, porque, ya sabe, parece que los datos tienen una muy alta probabilidad de estar en la región central, baja probabilidad de estar lejos de la misma, baja probabilidad de estar más lejos y así sucesivamente. Quizá esta sea una estimación razonable de «Mu» y de «sigma» al cuadrado, que sí corresponde a una distribución Gaussiana, entonces luego tendrá esta apariencia.   
  
Entonces, lo que haré es escribir las fórmulas, la fórmulas estándar para para estimar los parámetros de «Mu» y «sigma» cuadrada. La forma en que vamos a estimar «Mu» será sólo el promedio de mi ejemplo. Entonces «Mu» es el parámetro medio, voy a tomar mi conjunto de entrenamiento, tomo mis ejemplos "m" y los promedio.



Esto sólo me da el centro de esta distribución pero ¿que hay de «sigma» al cuadrado? Bueno, la varianza, sólo voy a escribir la fórmula estándar de nuevo, voy a estimarla como



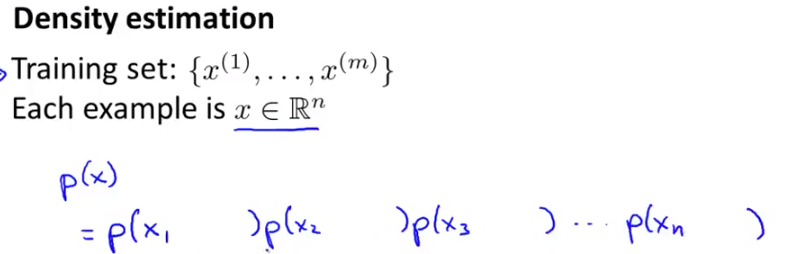
Lo que es la varianza o una interpretación de la varianza, es que si se fija en este término, es la diferencia al cuadrado entre el valor que obtuve en mi ejemplo, menos la media, menos el centro, menos la media de distribución. Como sabe, la varianza, la voy a estimar como sólo el promedio de las diferencias al cuadrado, entre mis ejemplos, menos de la media.   
  
Sólo como un comentario adicional, sólo para aquellos que son expertos en las estadísticas, si es un experto en estadística y si ha escuchado de la estimación máxima de verosimilitud (*maximum likehood estimation*), entonces, estas estimaciones son en realidad la estimación de la probabilidad máxima de los parámetros «Mu» y «sigma» cuadrada. Pero si no ha oído sobre eso antes, no se preocupe, todo lo que necesita saber es que estas son las dos fórmulas estándar para tratar de averiguar lo que «Mu» y «sigma» cuadrada valen, dado el conjunto de datos.   
  
Finalmente, una última observación: de nuevo, sólo para aquellos que tal vez han tomado una clase de estadísticas antes. Si usted ha tomado una clase de estadística antes, algunos de ustedes puede que hayan visto la fórmula aquí, donde, ya sabe, esta es "m" menos 1, en vez de "m" siendo así la fórmula de la *cuasi-varianza*. De modo que este primer término se convierte en 1 sobre m-1, en lugar de 1 sobre m. En el aprendizaje automático, las personas tienden a usar esta fórmula de 1 sobre m pero en la práctica, el hecho de que sea 1 sobre m o 1 sobre m-1, no hace ninguna diferencia esencial, suponiendo que, m es razonablemente grande, es un conjunto de entrenamiento de gran tamaño. Así que, sólo en caso de que haya visto esta otra versión antes, en cualquier versión funciona igualmente bien pero en el aprendizaje automático, la mayoría de las personas tiende a usar 1 sobre m en esta fórmula.Las dos versiones tienen propiedades teóricas ligeramente diferentes, propiedades matemáticas un poco distintas pero en la práctica, realmente hace muy poca diferencia, si la hay.   
  
Pues bien, espero que ahora tengan una buena idea de cómo luce la distribución Gaussiana y cómo estimar los parámetros «Mu» y «sigma» al cuadrado de la distribución de Gaussiana y si le dan un conjunto de entrenamiento, esto es, si tiene un conjunto de datos que sospecha que proviene de una distribución Gaussiana con parámetros desconocidos utilizando «sigma» al cuadrado.

# Algoritmo

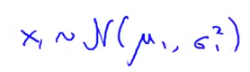
En el último vídeo hablamos acerca de la distribución Gaussiana. En este video aplicaremos eso para desarrollar un algoritmo de detección de anomalías

Digamos que tenemos un conjunto de “m” ejemplos de entrenamiento, y cada uno de estos ejemplos va a ser una función en Rn, por lo que su conjunto de entrenamiento podrían ser vectores de características de los últimos motores de aviones M que se fabrican. O podría ser características de m usuarios o algo así.

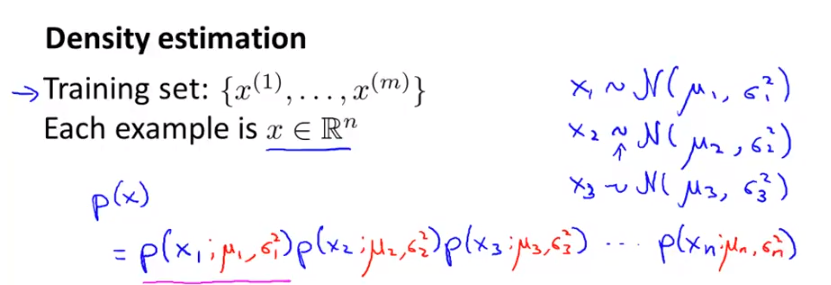
La forma en que vamos a tratar la detección de anomalías es que vamos a modelar p(x) del conjuntos de datos. Vamos a tratar de averiguar cuáles son las variables alta probabilidad, cuáles son tipos de variables de menor probabilidad. Entonces, "x" es un vector y lo que vamos a hacer es modelar p(x), como probabilidad de x, que es del primer componente de "x", por la probabilidad de x2 que es la probabilidad de la segunda función por la probabilidad de la tercera variable, y así sucesivamente hasta la probabilidad de la función final de xn.



Ahora voy a dejar espacio aquí porque voy a escribir algo en un minuto. Entonces, ¿cómo modelamos cada uno de estos términos, p(x1), p(x2) y así sucesivamente?   
  
Lo que haremos será asumir que la función, x1, se distribuye según una distribución Gaussiana, con cierta media «Mu» 1 y cierta varianza, que escribiré como «sigma» cuadrada 1,

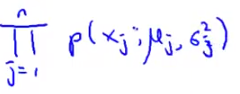


Entonces p(x1) va a ser una distribución de probabilidad Gaussiana, con "mi 1" media y varianza «sigma» al cuadrado 1. De manera similar, voy a asumir que, x2 es distribuida según el modelo Gaussiano, eso es lo que representa esta pequeña tilde, significa distribución según el modelo Gaussiano con media «Mu» 2 y ««sigma»» al cuadrado 2, así que se distribuye de acuerdo con una Gaussiana diferente, que tiene un conjunto diferente de parámetros, «Mu» 2, «sigma» al cuadrado2. Y, del mismo modo, ya sabe, x3 es todavía otra Gaussiana, así que esto puede tener una media diferente y una desviación estándar diferente a las otras variables y así sucesivamente, hasta xn. Y ese es mi modelo:

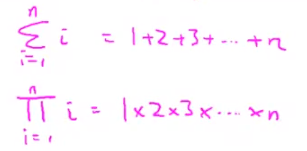


Al igual que un comentario lateral para aquellos de ustedes que son expertos en estadística, resulta que esta ecuación que acabo de escribir en realidad corresponde a una suposición de independencia sobre los valores de las características x1 a través de xn. Pero en la práctica resulta que el algoritmo de este fragmento funciona muy bien, independientemente de si estas características son independientes o no, e incluso si la suposición de independencia no es cierta, este algoritmo funciona bien. Pero en caso de que no conozca esos términos, solo utilicé suposiciones de independencia, etc., no se preocupe. Podrás entenderlo e implementar este algoritmo sin problemas y ese comentario fue realmente solo para los expertos en estadísticas.

Finalmente, con el fin de terminar, permítame tomar esta expresión y escribirla de forma un poco más reducida. Entonces, vamos a escribir que este producto desde j=1 hasta "n" de p(xj) parametrizada por "mi j", «sigma» cuadrada "j":

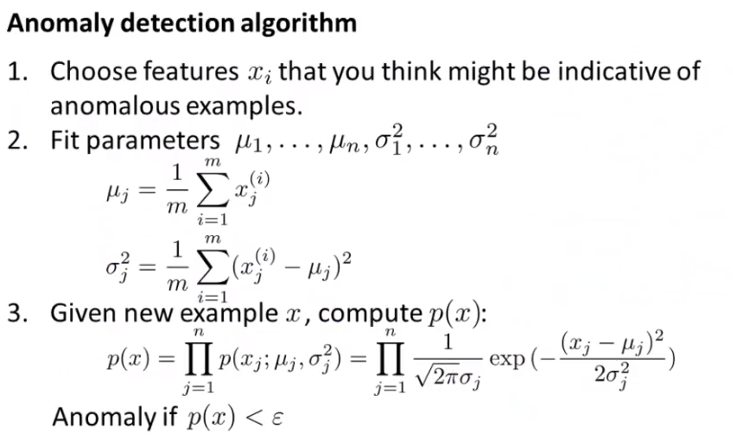


Este gracioso símbolo aquí, es la «pi» mayúscula del alfabeto griego, ese símbolo raro allí corresponde a tomar el producto de un conjunto de valores. Y así, se familiariza con la notación sumatoria, así que la suma de "i" igual a 1 a través de n, de "i". Esto significa 1 + 2 + 3 más punto, punto, punto, hasta "n". Donde este símbolo raro aquí, este símbolo de producto, el producto de i=1 a través de "n" de "i". Entonces esto significa que, es igual a la suma excepto que ahora estamos multiplicando. Esto se convierte en 1 x 2 x3 hasta "n":



Y al usar esta notación de producto, este producto de j= 1 a trav´´es de "n" de esta expresión es más compacto, esta es una manera más corta de escribir este producto de todos esos términos allá arriba. Puesto que estamos tomando esta p(xj), dados los términos "mi j", «sigma» cuadrada "j" y los multiplicamos juntos.

Por cierto, el problema de la estimación de esta distribución p(x), a veces se conoce como el problema de la estimación de la densidad. Por lo tanto, ese es el título de la diapositiva, al reunir todo esto, se forma un algoritmo de detección de anomalías.

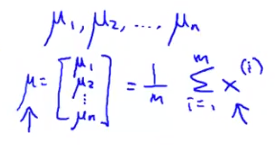
  
  
El primer paso es elegir variables o encontrar variables "xi"que pensamos que pueden ser indicativas de ejemplos anómalos. A lo que me refiero con eso es, tratar de encontrar variables, de modo que cuando haya un usuario inusual en su sistema que puede estar haciendo cosas fraudulentas, o cuando en el ejemplo del motor de avión, ya sabe, hay algo curioso, algo extraño en uno de los motores de avión, elija variables xi, que crea que podrían tomar inusualmente grandes valores, o inusualmente valores pequeños, que den la apariencia de un ejemplo anómalo.

De manera más general, sólo trate de elegir las variables que describen las propiedades generales de las cosas sobre las que está recolectando datos. A continuación, dado un sistema de entrenamiento de "m", ejemplos con valores no asignados, x1 a xm.

Luego ajuste los parámetros, «Mu» 1 a «Mu» n y «sigma» cuadrada 1 a «sigma» cuadrada "n", así que estas fórmulas fueron similares a las fórmulas que teníamos en el video anterior, que íbamos a usar para estimar cada uno de estos parámetros y para dar una interpretación,«Mu» "j" que es mi valor promedio de la variable "j", «Mu» "j" entra en este término p(xj), parametrizado por «Mu» "j" y «sigma» cuadrada "j".

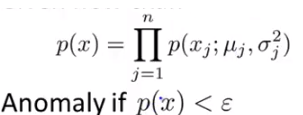


Esto indica que "mi j" sólo toma la media sobre mi conjunto de valores de entrenamiento de la variable "j". Y, sólo por mencionar, que hace esto, usted calcula estas fórmulas para j=1 hasta "n". Así que use estas fórmulas para estimar «Mu» 1, para estimar «Mu» 2 y así sucesivamente hasta «Mu» n y de forma similar para «sigma» cuadrada, y también es es posible inventar versiones vectorizadas de éstos. Así es que si piensa en «Mu» como un vector, entonces, si «Mu» es un vector, tenemos «Mu» 1, «Mu» 2, hasta "mi n"

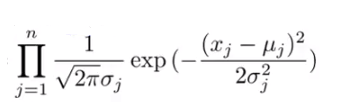


Después se puede escribir una versión vectorizada de ese conjunto de parámetros, de este modo.La suma de i=1 por medio de n, x(i). Esta fórmula que acabo de escribir estima esta "xi" como los vectores de dirección que estiman «Mu» para todos los valores de "n" de forma simultánea. También es posible encontrar una fórmula vectorizada para estimar «sigma» cuadrada "j".

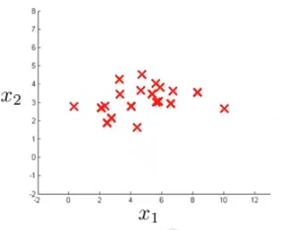
Por último, cuando le den un nuevo ejemplo, cuando tenga un nuevo motor de avión y quiera saber si el motor de este avión es anómalo, lo que tenemos que hacer es calcular p(x), que es la probabilidad de este nuevo ejemplo. Entonces, p(x) es igual a este producto:



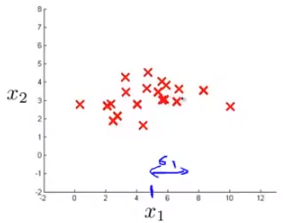
Y lo que usted implementa, lo que calcula es esta fórmula:



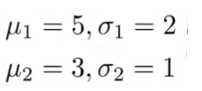
que es sólo la fórmula para la probabilidad Gaussiana, así usted calcular esto y finalmente si esta probabilidad es muy pequeña, entonces marca esto como una anomalía.   
  
Aquí hay un ejemplo de una aplicación de este método.   
  
Digamos que tenemos este conjunto de datos:



Si se fija, bueno, vamos a ver la función de x1. si observa este grupo de datos, parece que en promedio, las variables x1 tienen una media de aproximadamente 5 y la desviación estándar, si sólo mira los valores de x1 de este conjunto de datos tiene una desviación estándar de quizá 2. Así que esa es «sigma» 1.

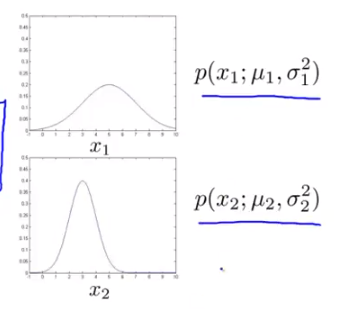


Parece que x2, ya sabe, los valores de las las variables, tal como se miden en el eje vertical, pareciera que tiene un valor promedio de aproximadamente 3, y una desviación estándar de aproximadamente 1. Si toma este conjunto de datos y si estima "mi 1", mi 2", «sigma» 1, «sigma» 2 esto es lo que obtiene.

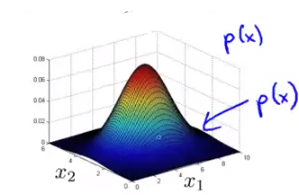


Y otra vez, estoy escribiendo «sigma» aquí, estoy pensando en desviaciones estándar, pero la fórmula de la diapositiva anterior, en realidad dio las estimaciones de los cuadrados de las cosas: «sigma» cuadrada 1 y «sigma» cuadrada 2. Sólo tenga cuidado si está usando «sigma» 1, «sigma» 2, «sigma» cuadrada 1 o «sigma» cuadrada 2. Entonces, «sigma» cuadrada 1, por supuesto, sería igual a 4, por ejemplo, como el cuadrado de 2.

En imágenes se muestra que p(x1) parametrizada por «Mu» 1 y «sigma» al cuadrado 1, y p(x2) parametrizada por «Mu» 2 y «sigma» cuadrada 2, luciría como estas dos distribuciones aquí.



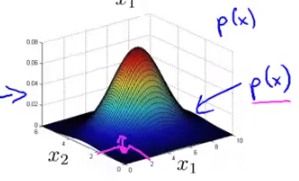
Resulta que si tuviéramos que trazar p(x), que es el producto de estas dos distribuciones, en realidad puede obtener un gráfico de superficie que tiene este aspecto.



Este es un trazo de p(x) donde la altura por encima de este, donde la altura de esta superficie en un punto particular, dados los valores en particular de x1 y x2 tenemos la probabilidad dada por la altura de esta superficie 3D.

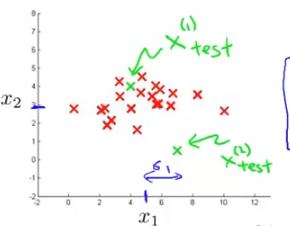
Así que p(x), que es la altura de este diagrama, es literalmente p(x1) parametrizada por «Mu»12, «sigma» al cuadrado 1, multiplicada por p(x2) parametrizada por «Mu» 2 «sigma» cuadrada 2.

Así es como ajustamos los parámetros a estos datos. Vamos a ver si tenemos un par de ejemplos nuevos, tal vez tengo un nuevo ejemplo aquí:

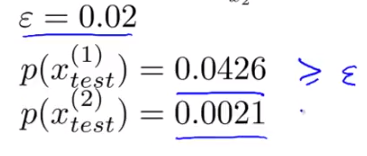


Donde x1 = 2 y x2 = 2. ¿Es una anomalía o no? o, tal vez tenga un ejemplo distinto, quizá tenga un segundo ejemplo diferente allí. Entonces, ¿es una anomalía o no?

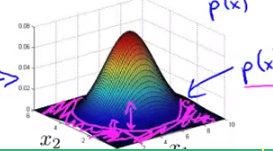
La forma en que hacemos esto, es que estableceríamos un valor para «épsilon», vamos decir que he elegido «épsilon»= 0.02. Diré más adelante cómo elegimos «épsilon» pero vamos a tomar este primer ejemplo, déjeme nombrar esto como ejemplo "x1 prueba" y nombraré al segundo ejemplo como "x2 prueba".



Lo que hacemos es calcular p(x1) de "x1 prueba", así que utilizamos esta fórmula para calcularlo y esto parece un valor muy grande. En particular, esto es mayor o igual a «épsilon» y entonces esta es una muy alta probabilidad, al menos más grande que «épsilon», entonces diremos "x1 prueba" no es una anomalía. Considerando que, si usted calcula p(x2 prueba) bueno, es sólo un valor mucho menor. Entonces si esto es menor a épsilon, diremos que de hecho si es una anomalía porque es mucho menor al «épsilon» que elegimos.



Y de hecho, podría mejorarlo aquí, lo que realmente muestra esto es que, si se fija en el diagrama de superficie 3D, está indicando que todos los valores de x1 y x2 que tienen una gran altura por encima de la superficie, corresponden a un ejemplo no anómalo, a un ejemplo correcto o normal. Mientras que todos los puntos lejos de aquí, todos los puntos fuera tienen una probabilidad muy baja, entonces vamos a marcar esos puntos como anómalos, y así se va a definir alguna region, que tal vez se vea así, de modo que todo lo que está fuera de esto, lo marca como anómalo.





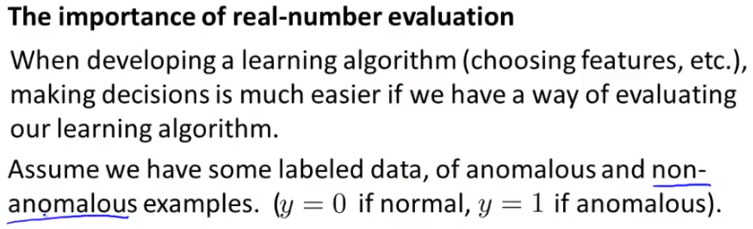
Mientras que las cosas dentro de esta elipse que acabo de dibujar, se consideran ejemplos normales, no anómalos o no anormales. Entonces este ejemplo de "x2 prueba" está fuera de esa región, por lo que tiene una probabilidad muy pequeña, entonces lo consideramos un ejemplo anómalo.

En este vídeo hablamos acerca de cómo estimar p(x), la probabilidad de x, con el propósito de desarrollar un algoritmo de detección de anomalías. En este vídeo también nos adentramos en un proceso completo para obtener conjuntos de datos, estuvimos ajustando los parámetros, haciendo cálculos de parámetros. Obtuvimos los parámetros «Mu» y «sigma», y tomamos nuevos ejemplos para decidir si los nuevos ejemplos eran anómalos o no.   
  
En los próximos videos seguiremos profundizando en este algoritmo y hablaré un poco más acerca de cómo hacer que en realidad funcione bien.

# Desarrollando y evaluando un sistema de detección de anomalías

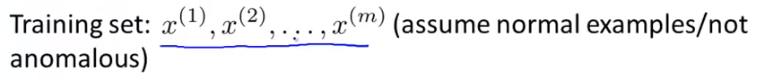
En el último video, desarrollamos un algoritmo de detección de anomalías. En este video, me gustaría hablar sobre el proceso de cómo proceder para desarrollar una aplicación específica de detección de anomalías a un problema y en particular esto se centrará en el problema de cómo evaluar un algoritmo de detección de anomalías. En videos anteriores, ya hemos hablado sobre la importancia de la evaluación de números reales y esto capta la idea de que cuando usted está tratando de desarrollar un algoritmo de aprendizaje para una aplicación específica, necesita a menudo hacer muchas elecciones tales como, por ejemplo, elegir qué variables usar y así sucesivamente. Y tomar decisiones sobre todas estas opciones es a menudo mucho más fácil si tiene una forma de evaluar su algoritmo de aprendizaje que de como resultado un sólo número.   
Pero si está tratando de decidir, ya sabe, tengo una idea para una variable adicional, ¿debo incluir esta variable o no? Si puede ejecutar el algoritmo con la variable y ejecutar el algoritmo sin la variable, sólo obteniendo como resultado un número que le indica, ya sabe, si mejoró o empeoró el desempeño al añadir esta variable. Entonces le da una mucho mejor manera, una manera mucho más simple, con la cual decidir si desea o no incluir esa variable. Así que con el fin de ser capaz de desarrollar de forma rápida un sistema de detección anomalía, sería muy útil contar con una manera de evaluar un sistema de detección de anomalías.   
  
Para hacer esto, con el fin de evaluar un sistema de detección de anomalías, en realidad vamos a asumir que tenemos algunos datos con valores asignados. Así que, hasta ahora, estaremos considerando a la detección de anomalías como un problema de aprendizaje no supervisado, que usa datos sin valores asignados pero si tiene algunos datos con valores asignados que especifiquen cuáles son algunos ejemplos anómalos y cuáles son algunos ejemplos no anómalos, entonces, así es realmente como pensamos que es la manera estándar de evaluar un algoritmo de detección de anomalías.

Así que, retomando el ejemplo del motor de avión, digamos que, tenemos algunos datos con valores asignados de unos pocos ejemplos anómalos de algunos motores de avión que fueron fabricados en el pasado y que resultaron ser anómalos. Resultó que eran deficientes o extraños de alguna manera, digamos que también tenemos algunos ejemplos normales o no anómalos, algunos ejemplos perfectamente buenos. Voy a utilizar y= 0 para denotar al ejemplo normal o no anómalo y y=1 para denotar los ejemplos anómalos.



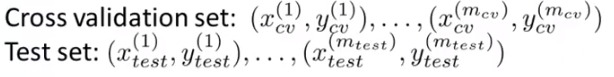
El proceso de desarrollar y evaluar un algoritmo de detección de anomalías es el siguiente:

Vamos a pensar en esto como un grupo de de entrenamiento y vamos a hablar acerca de la validación cruzada en grupos de prueba más tarde pero con frecuencia pensamos en el grupo de entrenamiento como un grupo de datos que aún no tiene valores asignados. Así es que, tenemos una gran colección de ejemplos normales, no anómalos, no anormales.

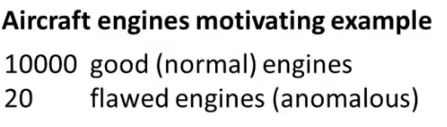


Y generalmente consideramos esto como no anómalo pero en realidad está bien, incluso si algunas anomalías se infiltraron en su conjunto de entrenamiento con valores no asignados.

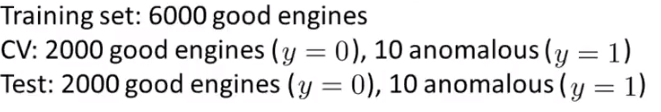
A continuación, vamos a definir un conjunto de validación cruzada y un conjunto de prueba, con el cual evaluar un algoritmo de detección de anomalías.



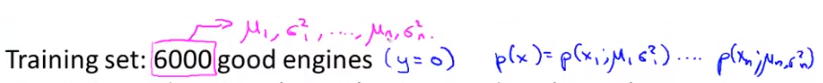
Así que, específicamente, para ambos grupos de prueba de validación cruzada, asumimos que, ya sabe, podemos incluir algunos ejemplos en el conjunto de validación cruzada y conjunto de prueba que contienen ejemplos que son conocidos por ser anómalos. De modo que los conjuntos de prueba, digamos que tenemos unos cuantos ejemplos con y=1 que corresponden a los motores de avión anómalos.   
  
Aquí tenemos un ejemplo concreto:



Digamos que, en conjunto, estos son los datos que tenemos, hemos fabricado 10,000 ejemplos de motores que, hasta donde sabemos son motores de avión perfectamente normales, elaborados perfectamente. Y una vez más, está bien incluso si algún motor deficiente se infiltra en el conjunto de 10, 000, en realidad está bien pero de algún modo supusimos que la gran la mayoría de estos 10,000 ejemplos son, ya sabe, motores buenos y normales, no anómalos. Y digamos que, históricamente, sin importar cuánto tiempo hemos estado trabajando en la planta de fabricación, digamos que terminamos obteniendo las variables, de 24 a 28 motores anómalos también. Un uso muy típico de la detección de anomalías, ya sabe, los ejemplos de número no anómalos, esto es, con y=1, podemos tener, por ejemplo, 20 a 50. Sería un rango de ejemplos muy típico, un número de ejemplos que tenemos con y=1. Y, por lo general, tendremos un número mucho mayor de buenos ejemplos.   
Con estos datos, una forma bastante típica de dividirlos entre el conjunto de entrenamiento, el conjunto de validación y el conjunto de prueba, sería esta:

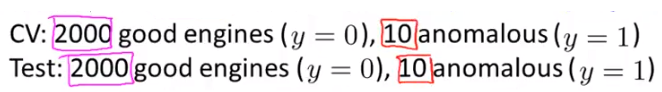


Vamos a tomar 10,000 motores de avión buenos y ponemos 6,000 de estos en el conjunto de entrenamiento sin valores asignados. Entonces, denomino esto como un grupo de entrenamiento sin valores asignados pero en todos estos ejemplos son los que realmente corresponden a y=0, hasta donde sabemos, y así, vamos a utilizar esto para ajustar p(x), ¿verdad? Así que, vamos a utilizar estos 6,000 motores para ajustar p(x), que es la p(x)1 parametrizada por "mi 1", «sigma» al cuadrado 1, hasta p(xn) parametrizada por «Mu» n, «sigma» al "n".

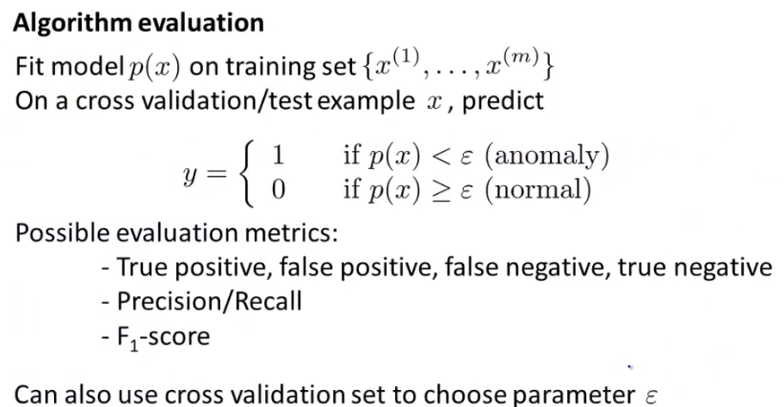


Estos serían los 6,000 ejemplos que queremos utilizar para estimar los parámetros: letra griega «Mu»1, «sigma» cuadrada 1, hasta mi "n", «sigma» cuadrada "n", entonces, ese es nuestro conjunto de entrenamiento de todos, ya sabe, buenos ejemplos o la gran mayoría de buenos ejemplos.

A continuación, tomaremos nuestros motores de avión buenos y vamos a colocar algunos de ellos en un grupo de validación cruzada, más algunos otros de ellos en los conjuntos de prueba. Entonces, 6,000 más 2,000 más 2,000, así es cómo separamos nuestros 10 mil motores de avión elaborados perfectamente. Posteriormente, también tenemos 20 motores de aviones defectuosos, los tomaremos y tal vez los dividiremos para poner diez de ellos en el conjunto de validación cruzada y otros diez en los conjuntos de prueba. En la siguiente diapositiva, hablaremos sobre cómo usar en realidad esto para evaluar un algoritmo de detección de anomalías.

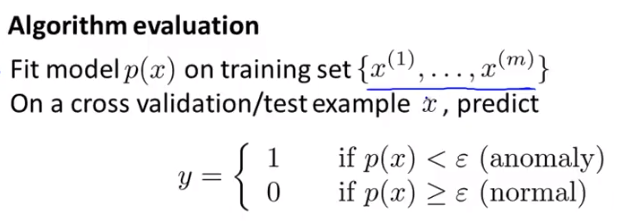


Bien, lo que ha descrito aquí es, cómo decirlo, probablemente la recomendación de una buena manera de dividir el ejemplo asignación de valores y sin valores sin asignar de los motores de avión buenos y los defectuosos, donde utilizamos una separación de 60, 20, 20% para los motores buenos y tomamos los motores defectuosos y los colocamos en el grupo de validación cruzada y sólo en el conjunto de prueba. Luego, en la siguiente diapositiva veremos porqué se da este caso.   
  
Como nota, si usted mira cómo la gente aplica los algoritmos de detección de anomalías, a veces ve que otras personas dividen los datos de forma distinta también. Otra alternativa, esta no es en realidad una alternativa recomendable pero algunas personas quieren quitar sus 10,000 motores buenos, tal vez poner 6,000 de ellos en su conjunto de entrenamiento y y después poner los mismos 4,000 en el grupo de validación cruzada y en el conjunto de pruebas. Y entonces, ya sabe, nos gusta pensar que el conjunto de validación cruzada y el conjunto de prueba son conjuntos de datos totalmente diferentes entre sí. Pero sabe, en la detección de anomalías, a veces vemos personas que de cierto modo usan el mismo conjunto de motores buenos en el conjunto de validación cruzada y en los conjuntos de prueba y a veces ve gente que usa exactamente los mismos conjuntos de motores anómalos en el conjunto de validación cruzada y en el conjunto de prueba. Y así, todas estas son consideradas, ya sabe, prácticas menos adecuadas y definitivamente menos recomendables. Ciertamente, usando los mismos datos en el grupo de validación cruzada y en el conjunto de pruebas, no se considera una buena práctica de aprendizaje automático. Pero, a veces ve gente que también hace este tipo de cosas.   
  
Dados los conjuntos de entrenamiento, de validación cruzada y de prueba, así es cómo se evalúa o cómo se desarrolla y evalúa un algoritmo:

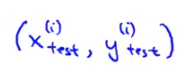


Primero, tomamos los conjuntos de entrenamiento y ajustamos el modelo p(x). Entonces, ajustamos así, todas estas Gaussianas a mis ejemplos de valores no asignados a "m" de motores de avión anómalos y a estos, los llamo ejemplos con valores no asignados pero en realidad son ejemplos que estamos asumiendo como ejemplos buenos, son los motores de avión normales.   
  
Luego, imagine que su algoritmo de detección de anomalías en realidad está haciendo una predicción, entonces en la validación cruzada del grupo de prueba dado el, digamos, ejemplo de prueba "x", piense que el algoritmo está prediciendo que y=1, p(x) es menor que «épsilon», debemos estar tomando cero, si p(x) es mayor o igual a «épsilon». Así, dada x, está tratando de predecir, cuál es el valor asignado, dada "y= 1",que corresponde a una anomalía o ¿será que y= 0 corresponde a un ejemplo normal?

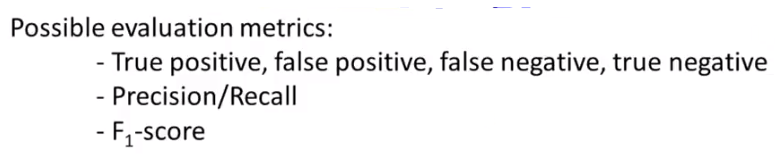
Así que Dados los conjuntos de entrenamiento, validación cruzada y de prueba, ¿cómo desarrolla un algoritmo? Y más específicamente, ¿cómo evalúa un algoritmo de detección de anomalías? Bueno, a todo esto, el primer paso es tomar el conjunto de entrenamiento valor no asignado etiquetado y ajustar el modelo p(x) para guiar los datos de entrenamiento. Así que toma el conjunto de entrenamiento con valores no asignados pero realmente, estos son ejemplos que suponemos, la mayoría de los cuales son los motores de avión normales, no porque no sean anómalos y esto ajustará el modelo p(x). Ajustará todos aquellos parámetros para todas las Gaussianas en los datos.



Después, en la validación cruzada de este conjunto de prueba, vamos a pensar que el algoritmo de detección de anomalías está tratando de predecir el valor de "y". Así que en cada uno de, digamos, los ejemplos de prueba, tenemos estas "x1 prueba", "y1 prueba"



Donde "y" será igual a 1 o 0, dependiendo de si este era un ejemplo anómalo.   
  
Así que dada la entrada "x" en mi conjunto de prueba, pienso que mi algoritmo de detección de anomalías esta prediciendo la "y" como 1 si p(x) es menor que «épsilon». Así que la probabilidad de predecir que esta es una anomalía, es muy baja. Por el contrario, pensamos que el algoritmo está prediciendo que y es igual a 0. Si p(x) es mayor o igual a «épsilon», entonces la predicción de esos ejemplos normales en la p(x) es razonablemente grande.   
  
Así que ahora podemos pensar que el algoritmo de detección de anomalías está realizando predicciones para saber cuáles son los valores de estas etiquetas "y" en los conjuntos de prueba o en el conjunto de validación cruzada y esto nos pone de forma más similar al entorno de aprendizaje supervisado, ¿verdad? en donde tenemos el conjunto de prueba con valores asignados y nuestro algoritmo está haciendo predicciones sobre estos valores asignados y así podemos evaluarl qué tan a menudo obtiene correctamente estos valores asignados. Por supuesto, estos valores asignados serán muy sesgadas porque y=0, que representa los ejemplos normales, suele ser mucho más común que y=1, que representa los ejemplos anómalos. Pero, como sabe, esto está mucho más cerca a la fuente de métrica de evaluación que podemos usar en el aprendizaje supervisado.   
  
Entonces ¿cuál es una buena métrica de evaluación para utilizar?

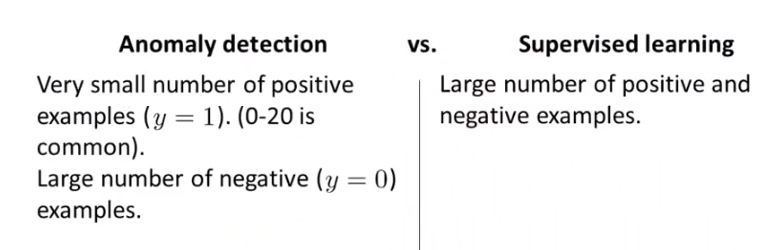


Bueno, debido a que los datos son muy sesgados porque y=0 es mucho más común, la exactitud de la clasificación no sería una buena métrica de evaluación, ya habíamos hablado de esto en el video anterior. Por lo que si tiene conjuntos de datos muy sesgados, entonces predecir y=0 todo el tiempo tendrá una precisión de clasificación muy alta. Por el contrario, deberíamos usar una métrica de evaluación como el cálculo de la fracción de positivos verdaderos, falsos positivos, falsos negativos, verdaderos negativos o calcular la Precision/recall de este algoritmo o hacer cosas como calcular el puntaje F1, que es un único número real, para resumir la precisión y la recall. Así que éstas serían las maneras de evaluar un algoritmo de detección de anomalías en su conjunto de validación o en su conjunto de prueba.   
  
Finalmente, al inicio del algoritmo de detección de anomalías, también teníamos este parámetro «épsilon», ¿no? entonces, «épsilon» es este umbral que usaríamos para decidir cuándo marcar algo como una anomalía. Así que si tiene un conjunto de validación cruzada, otra forma de elegir este parámetro «épsilon» sería tratar con, probar con muchos valores diferentes de «épsilon», y luego elegir el del valor de «épsilon», que, digamos, maximiza el puntaje F1 o que de otro modo, funciona bien en su conjunto de validación cruzada.

# De manera más general, la forma de reducir los conjuntos de entrenamiento, de prueba y de validación cruzada es que cuando estamos tratando de tomar decisiones, como qué variables incluir o tratamos de, por ejemplo, ajustar el parámetro «épsilon», entonces tendríamos que evaluar continuamente el algoritmo en los conjuntos de validación cruzada y tomar todas esas decisiones, tales como qué variables uso, cómo ajustar «épsilon», usar esto para evaluar el algoritmo en el conjunto de validación cruzada y luego cuando hayamos elegido el conjunto de variables, cuando hayamos encontrado el valor de «épsilon» con el que estemos satisfechos, entonces podemos tomar el modelo final y evaluarlo, ya sabe, hacer la evaluación final del algoritmo en los conjuntos de prueba. En este vídeo hablamos acerca del proceso de cómo evaluar un algoritmo de detección de anomalías y de nuevo, habiendo sido capaces de evaluar un algoritmo, con una evaluación de un solo número real con un número, tal como el puntaje de F1, que a menudo le permite usar de manera más eficiente manera de invertir tu tiempo de desarrollar un sistema de detección de anomalías, cuando tratamos de tomar este tipo de decisiones, si tengo que elegir «épsilon», qué variables voy incluir y cosas por el estilo. En este video, empezamos a utilizar un poco de datos valores asignados, con el fin de evaluar el algoritmo de detección de anomalías y esto nos acerca un poco más a un entorno de aprendizaje supervisado. En el siguiente video, voy a decir un poco más sobre eso y en particular, vamos a hablar acerca de cuándo debe debe usar un algoritmo de detección de anomalías y cuando deberíamos estar considerando el uso del aprendizaje supervisado en su lugar, y cuáles son las diferencias entre estos dos formalismos.

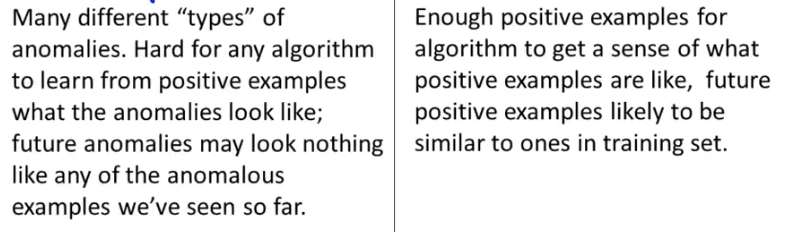
# Detección de anomalías vs aprendizaje supervisado

En el último vídeo hablamos sobre el proceso de cómo evaluar un algoritmo de detección de anomalías y allí empezamos a utilizar algunos datos con valores asignados, con ejemplos que sabíamos que eran anómalos o no anómalos, con y=1 o y=0.   
  
Entonces surge la pregunta, si tenemos estos datos valores asignados, tenemos algunos ejemplos que son conocidos por ser anomalías y algunos que no se conocen como anomalías, ¿por qué no simplemente usamos un algoritmo de aprendizaje supervisado? Así que ¿por qué no lo usamos la regresión logística o una red neuronal para tratar de aprender directamente de nuestros datos con valores asignados a predecir si y=1 o y=0. En este video, voy a tratar de compartir con usted algunas ideas y algunas pautas para saber cuándo probablemente debe usar un algoritmo de detección de anomalías y cuándo tal vez sea más fácil considerar el uso del algoritmo de aprendizaje supervisado.   
  
Esta diapositiva muestra cuáles son los casos bajo los cuales debe usar la detección de anomalías y cuándo podría ser más provechoso usar el aprendizaje supervisado.



Si usted tiene un problema con un número muy reducido de ejemplos positivos y recuerda que los ejemplos de y=1, son los ejemplos anómalos, entonces usted podría considerar usar un un algoritmo de detección de anomalía. Así que teniendo 0 a 20, tal vez hasta 50 ejemplos positivos, sería muy típico y por lo general, tenemos un pequeño conjunto de ejemplos positivos, vamos a reservar los ejemplos positivos sólo para los conjuntos de validación cruzada y de prueba. En contraste, en un típico caso normal de detección de anomalía, a menudo tenemos un número relativamente grande de ejemplos negativos, de estos ejemplos normales de motores de avión normales. Entonces podemos usar este número grande de ejemplos negativos, para ingresarlos al modelo p(x). Y así, hay esta idea en muchas aplicaciones de detección de anomalías, usted tiene muy pocos ejemplos positivos y muchos ejemplos negativos y cuando realizamos el proceso de estimación de p(x), de ajustar todos los parámetros Gaussianos, necesitamos sólo ejemplos negativos para hacer eso. Así que si usted tiene un montón de datos negativos, aún podemos ajustarlos muy bien a p(x).   
  
En cambio, para el aprendizaje supervisado, más típicamente tendríamos un número bastante grande de ejemplos tanto negativos, como positivos. Esta es sólo una manera de mirar el problema y decidir si debe usar un algoritmo de detección de anomalía o un algoritmo de aprendizaje supervisado.

Aquí hay otra forma en la que a menudo la gente piensa sobre los algoritmos de detección de anomalías.



Para aplicaciones de detección de anomalías con frecuencia hay muchos tipos diferentes de anormalidades, entonces piense en los motores de avión. Como sabe, hay muchas maneras diferentes en que los motores de avión pueden fallar ¿correcto? Hay tantas cosas que pueden salir mal, que podrían romper el motor de un avión. Por lo que, si ese es el caso y usted tiene un conjunto muy pequeño de ejemplos positivos, entonces puede ser difícil que un algoritmo aprenda a partir de su pequeño conjunto de ejemplos positivos, cómo lucen las anomalías. En particular, futuras anomalías pueden verse muy distintas a las que ha visto hasta ahora. Así que quizás en su conjunto de ejemplos positivos, haya visto 5, 10 o 20 diferentes formas en que puede fallar el motor de un avión. Pero tal vez mañana, necesite detectar un conjunto completamente nuevo, una forma de anomalía totalmente nueva, una nueva forma en que el motor de un avión puede dañarse y que nunca ha visto antes, si este es el caso, entonces, sería más prometedor sólo modelar los ejemplos negativos, con un tipo de modelo Gaussiano p(x), en vez de probar intensivamente de modelar los ejemplos positivos porque, como sabe, la anomalía de mañana puede ser muy distintas a las que ha visto hasta ahora.

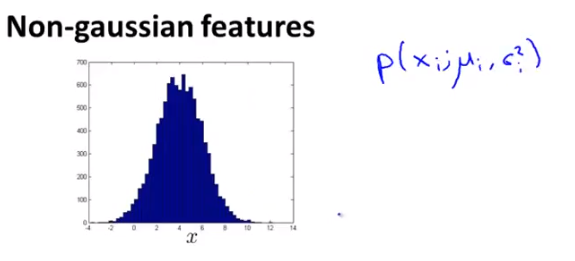
En contraste, en algunos otros problemas, tiene suficientes ejemplos positivos para que un algoritmo tenga una idea de cómo son los ejemplos positivos. Y, en particular, si piensa que probablemente los ejemplos positivos futuros serán similares a los del del conjunto de entrenamiento, entonces en ese caso, podría ser más razonable tener un algoritmo de aprendizaje supervisado, que se está rodeado de muchos ejemplos positivos, que puede mirar muchos ejemplos negativos, y usar esto para intentar distinguir entre ejemplos positivos y negativos.   
  
Espero que esto le haya dado una idea de, si tiene un problema específico y tiene que pensar sobre el uso del algoritmo de detección de anomalías o un algoritmo de aprendizaje supervisado. La diferencia clave en realidad es que en la detección de anomalías, después de tener un pequeño conjunto de ejemplos positivos con los que es imposible que un algoritmo de aprendizaje comprenda mucho a partir de ellos. Así que lo que podemos hacer en vez de esto es tomar un gran conjunto de ejemplos negativos y poder aprender mucho, comprender p(x) a partir de los ejemplos negativos de los motores de avión normales, digamos. Y reservamos el pequeño número de ejemplos positivos para evaluar nuestro algoritmo, para utilizarlo en los conjuntos de validación cruzada o de prueba.   
  
Sólo como un comentario sobre estos tipos diferentes de anomalías, ya sabe, en videos anteriores hablamos sobre los ejemplos de SPAM de correo electrónico. En esos ejemplos, hay en realidad muchos tipos diferentes de correo no deseado. El SPAM de correo electrónico está intentando venderle cosas por correo, tratando de robar sus contraseñas, esto se denomina “fishing” de correos y hay muchos tipos diferentes de SPAM. Pero para el problema del SPAM, por lo general tenemos bastantes ejemplos que ver, ya sabe, la mayoría de estos diferentes tipos de correos SPAM, porque tenemos un gran conjunto de ejemplos de SPAM, y por eso con frecuencia pensamos en el SPAM como un entorno de aprendizaje supervisado, aún cuando puede haber muchos tipos diferentes de SPAM.   
  
Si miramos algunas aplicaciones de detección de anomalías versus aprendizaje supervisado, encontraremos que:



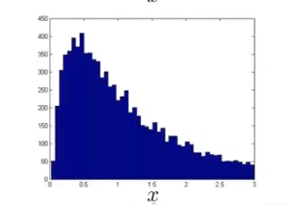
En la detección de fraudes, si tiene muchos tipos diferentes de de formas en que la gente trata de cometer un fraude y un conjunto de entrenamiento muy pequeño, un número reducido de usuarios fraudulentos en su sitio web, entonces usaría un algoritmo de detección de anomalías.   
  
Debo decir que si usted tiene, si es un mayorista en línea y en realidad ha tenido un gran número de personas que intenta cometer fraude en su sitio web, en realidad ya tiene un gran número de ejemplos donde y=1, entonces, a veces la detección de fraude en realidad podría cambiar a la columna de aprendizaje supervisado. Pero si no ha visto muchos ejemplos de usuarios que hacen cosas extrañas en su sitio de Internet entonces, con más frecuencia, la detección de fraudes en realidad se trata como un un algoritmo de detección de anomalías, en vez de un algoritmo de aprendizaje supervisado.   
  
Para mencionar otros ejemplos, hemos hablado ya de la fabricación, espero que vea más ejemplos normales y no tantas anomalías. D nuevo, para algunos procesos de fabricación, si está fabricando volúmenes muy grandes y ha visto muchos malos ejemplos, tal vez la fabricación también podría cambiar a la columna de aprendizaje supervisado. Pero si no ha visto muchos malos ejemplos de los productos viejos, entonces haré esta detección de anomalías.   
En el monitoreo de máquinas en el centro de datos, de nuevo podemos aplicar tipos de argumentos similares.   
  
Mientras que, la clasificación de correo SPAM , la predicción meteorológica y la clasificación de canceres, si tiene números iguales de ejemplos positivos y negativos, muchos de ustedes tienen muchos ejemplos de sus casos positivos y negativos, ejemplos, entonces, tendemos a tratar todos estos problemas como aprendizaje supervisado.   
  
Ojalá esto le de una idea de cuáles son las propiedades de un problema de aprendizaje que podría causar que usted marcara esto como un problema de detección de anomalías versus un problema de aprendizaje supervisado. Para muchos de los problemas enfrentados por varias empresas de tecnología y de ese tipo, en realidad estamos en estos contextos, donde tenemos muy pocos o a veces ningún ejemplo de entrenamiento positivo, tal vez hay algunos otros tipos diferentes de anormalidades que nunca hayamos visto antes y para esos tipos de problemas, muy a menudo, que el algoritmo que usamos es un algoritmo de detección de anomalías.

# Elegir que características utilizar

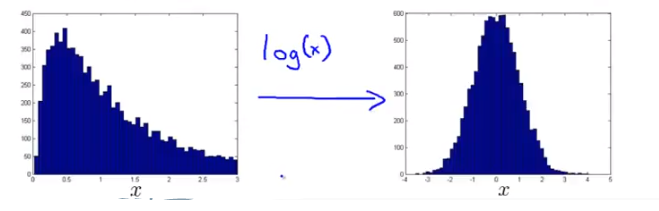
Para este momento, ya ha visto el algoritmo de detección y también hemos hablado de cómo evaluar un algoritmo de detección de anomalías Resulta que, cuando está aplicando una detección de anomalía, una de las cosas que tiene un enorme efecto sobre su buen funcionamiento es qué variables usted usa y qué variables elige para dar el algoritmo de detección de anomalías. En este video, lo que me gustaría hacer es decir algunas palabras, dar algunas sugerencias y pautas para saber cómo diseñar o seleccionar variables para dar a su algoritmo de detección de anomalías.   
  
En nuestro algoritmo de detección de anomalías, una de las cosas que hicimos fue modelar las funciones usando este tipo de distribución Gaussiana. Con xi, «Mu», «sigma» cuadrada i, por ejemplo.



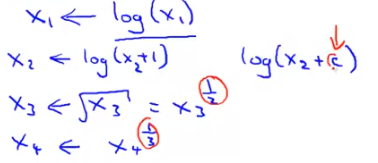
Y una cosa que a menudo hago es trazar los datos o el histograma de los datos, para asegurarme de que los datos parecen vagamente Gaussianos antes de alimentarlos a mi algoritmo de detección de anomalías. Y, generalmente podrá funcionar bien, aunque los datos no sean Gaussianos, sin embargo, se trata de un buen chequeo sanitario para ejecutar. Por cierto, en caso de que sus datos no parezcan Gaussianos, los algoritmos aún así funcionarán bien. En concreto, si trazo mis datos así y si se parecen a un histograma como este, y la manera de trazar un histograma es usar HIST, o el comando HIST en Octave pero así es como luce, parece vagamente Gaussiano, así que si mis variables lucen así:



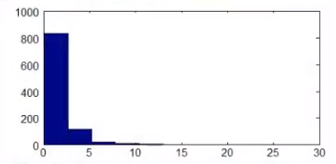
estaría muy feliz de alimentarlas a mi algoritmo pero si fuera a trazar un histograma de mis datos y luciera así, bueno, esto no se parece en absoluto a una curva en forma de campana, esta es una distribución muy asimétrica, tiene forma de pico a un lado.   
  
Si así es como se ven mis datos, lo que a menudo haré será jugar con diferentes transformaciones de los datos, con el fin de que luzcan más Gaussianos. Otra vez, el algoritmo generalmente funcionará bien, incluso si no hace esto pero si usa estas transformaciones para hacer sus datos más Gaussianos, podría funcionar un poco mejor.   
  
Así que tenemos el conjunto de datos que se ve así, lo que puedo hacer es tomar una transformación logarítmica de datos, si hago eso y vuelvo a trazar el histograma, lo que obtengo en este ejemplo particular, es un histograma que tiene este aspecto:



Y esto se ve mucho más Gaussiano, ¿verdad? se parece mucho más a la clásica curva en forma de campana, que podemos ajustar con algunos parámetros «sigma» de media y varianza. A lo que me refiero con una transformación logarítmica, es realmente, si tengo alguna variable x1 y luego el histograma de x1 se ve así, entonces podría tomar mi variable x1 y reemplazarla con el registro de x1 y esta es mi nueva x1 que podré trazar al histograma más a la derecha y esto parece mucho más Gaussiano.   
  
En lugar de sólo hacer una transformación logarítmica, otras cosas que puede hacer, podría ser, vamos a decir que tengo una variable diferente, x2, tal vez voy a reemplazar ese log(x+1) o más generalmente con log (x2 + c) y algunas constante c y esta constante puede ser algo con que jugar, para tratar de hacer que parezca lo más Gaussiano posible. O para otra variable x3, tal vez la reemplazaré con x3 que puede ser en raíz cuadrada. La raíz cuadrada es sólo x3 a la potencia de una mitad, ¿verdad? y esta mitad es otro ejemplo de un parámetro con el que puedo jugar. Entonces, tendría x4 y tal vez podría reemplazar en su lugar con x4 a otra potencia, tal vez a la potencia de 1/3. Y éstos, todos estos, este otro, este parámetro exponente, o el parámetro "c", todos estos son ejemplos de parámetros con los que puede jugar, con el fin de hacer parecer sus datos un poco más Gaussianos.



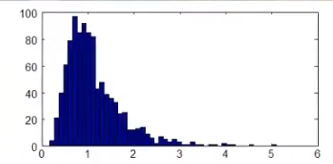
Déjeme darle una demostración en vivo de cómo jugar en realidad con mis datos para que parezcan más Gaussianos. Bien, ya lo cargué en Octave, aquí hay un conjunto de variables x, tengo mil ejemplos cargados allí. Vamos a trazar el histograma de mis datos, use el comando hist(x), y aquí está mi histograma:



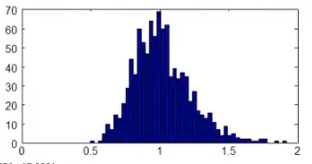
Por defecto, creo que esto utiliza 10 contenedores de histogramas, pero yo quiero ver un histograma de cuadrícula más fina. Así que activamos hist(x, 50), entonces, esto se traza en 50 contenedores diferentes.



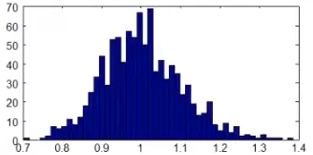
Muy bien, se ve mejor. Ahora, esto no parece muy Gaussiano, ¿verdad? entonces, vamos a empezar a jugar con los datos. Probemos con un hist (x.^0.5, 50), así que tomamos la raíz cuadrada de los datos y trazamos el histograma.



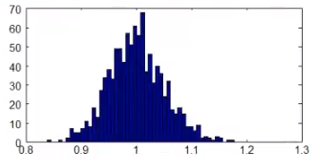
Y, bueno, parece un poco más de Gaussiano, pero no bastante allí, así que vamos a jugar en el parámetro 0.5. Veamos, si coloco esto a 0.2, se ve un poco más Gaussiano:



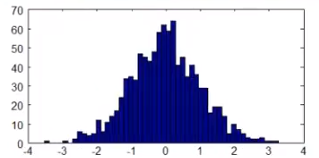
vamos reducir un poco más a 0.1.



Sí, se ve muy bien, en realidad podría usar 0.1, bueno, vamos a reducirla a 0.05

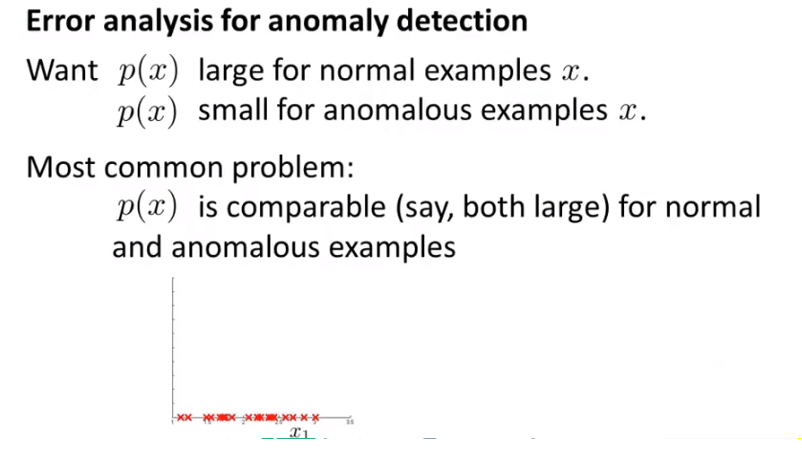


Y como puede ver esto parece bastante Gaussiano, entonces puedo definir una nueva variable que es "xnueva = x.^0.05" y ahora mi nueva función "xnueva" se ve más Gaussiana que la anterior y ahora podría usar en su lugar esta nueva variable para alimentar a mi algoritmo de detección de anomalías. Y por supuesto, hay más de una manera de hacer esto. También puede usar hist de log(x), que es otro ejemplo de una transformación que puede utilizar.

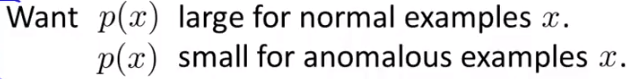


Y, sabe, eso también se ve muy Gaussiano, también puedo definir xnueva = log (x) y esa sería otra muy buena opción de una función que puedo usar.   
  
Para resumir, si traza un histograma con los datos y nota que no luce muy Gaussiano, vale la pena jugar un poco con diferentes transformaciones como estas, para ver si puede hacer que sus datos parezcan un poco más Gaussianos, antes de alimentarlos a su algoritmo de aprendizaje, aunque incluso si no lo hace, puede que funcione bien, pero normalmente realizo este paso.

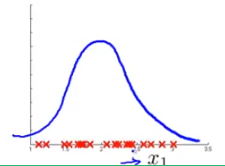
Lo segundo que quiero mencionar es cómo encontrar las variables para un algoritmo de detección de anomalías.   
  
La forma en que lo hago a menudo es a través de un procedimiento de análisis de error.



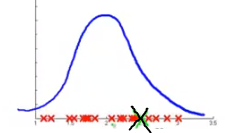
A lo que me refiero con eso es que es muy similar al procedimiento de análisis de error que tenemos para aprendizaje supervisado, donde entrenaría un algoritmo completo y le ejecutaría en un conjunto de validación cruzada, mire los ejemplos que fallaron y vea si se nos ocurren variables adicionales para ayudar al algoritmo a funcionar mejor en los ejemplo que fallaron en el conjunto de validación cruzada.   
  
Así que vamos a intentar razonar a través de un ejemplo de este proceso. En la detección de anomalías, esperamos que p(x) sea grande para los ejemplos normales y que sea pequeña para los ejemplos anómalos.



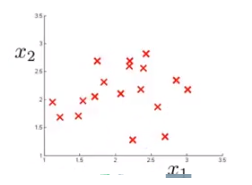
Un problema bastante común sería si p(x) es comparable, tal vez ambas son grandes para los ejemplos normales y los anómalos.   
  
Veamos un ejemplo concreto de ello: Digamos que estos son mis datos sin valores asignados, entonces aquí sólo tengo una variable, x1 y voy a aplicar una Gaussiana a esto. Puede ser que la Gaussiana que apliqué a mis datos se vea así



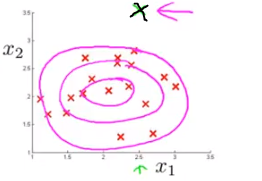
y ahora digamos que tengo un ejemplo anómalo, por decir, tengo un ejemplo anómalo que adquiere un valor x de 2.5. Entonces, trazo mi ejemplo anómalo allí y ya sabe, de cierta manera se entierra en medio de un grupo de ejemplos normales



y este simple ejemplo anómalo que he dibujado en negro, adquiere una probabilidad bastante alta, a la altura de la curva azul, y el algoritmo falla al marcar esto como un ejemplo anómalo.   
  
Ahora, si esto se tratara tal vez de la fabricación de motores de avión u otra cosa, lo que haría sería revisar mis ejemplos de entrenamiento y estudiar lo que salió mal con ese motor de avión particular, y al ver, si analizando este ejemplo puedo inspirarme a encontrar una nueva variable, x2, que ayude a distinguir entre este mal ejemplo, en comparación con el resto de mis ejemplos en rojo, en comparación con todos mis motores de avión normales. Si puedo hacer esto, la esperanza entonces sería que, si puedo crear una nueva variable, x2, cuando vuelva a trazar mis datos, si tomo todos mis ejemplos normales de mi conjunto de entrenamiento, quizá encuentre que todos mis ejemplos de entrenamiento son estas cruces rojas aquí.

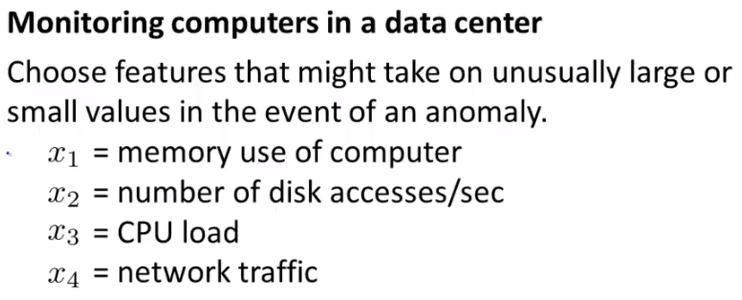


Y con suerte, si encuentro eso para mi ejemplo anómalo, la función x2 toma el valor inusual. Para mi ejemplo en color negro, esta anomalía, mi valor x1, sigue siendo 2.5. Luego, tal vez mi valor x2, con suerte adquiera un valor muy grande, tal como 3.5 en mi nueva variable, o un valor muy pequeño.   
  
Pero ahora, si modelo mis datos, encontraré que mi algoritmo de detección de anomalías da una alta probabilidad a los datos en las regiones centrales, una probabilidad ligeramente menor a eso, y probabilidad inferior a eso. Un ejemplo en esta parte exterior, sería que mi algoritmo ahora da una muy baja probabilidad en esta parte y así, el proceso de esto consiste en mirar realmente los errores que está cometiendo.



Mire la anomalía que el algoritmo está fallando en etiquetar y vea le ayuda a crear alguna nueva variable. Encuentre algo inusual sobre ese motor de avión y úselo para crear una nueva variable, con esta nueva variable resulta más fácil distinguir las anomalías de sus buenos ejemplos. Ese es el proceso de análisis de errores y su uso para crear nuevas variables para la detección de anomalías

Por último, permítame compartirle mi forma de pensar sobre cómo suelo elegir las variables para detección de anomalías. Con frecuencia, la forma en que concibo la elección de variables es que quiero elegir las variables que tomarán, ya sean valores muy muy grandes o muy muy pequeños, para ejemplos que creo que pueden resultar ser anomalías.   
  
Así que nuevamente vamos a usar nuestro ejemplo de la supervisión de equipos en un centro de datos. De modo que tiene varias máquinas, tal vez miles o decenas de miles de máquinas en un centro de datos y queremos saber si una de las máquinas, una de nuestras computadoras está fallando, haciendo algo extraño. Aquí están los ejemplos de las variables que usted puede elegir:



Quizá la memoria usada, el número de accesos de disco, la carga de CPU, el tráfico de red. Ahora, digamos que sospecho que uno de los casos de falla, digamos que en mi grupo de datos creo que la carga de CPU y el tráfico de red tienden a crecer linealmente uno con el otro. Tal vez estoy corriendo muchos servidores web y de ser así, si uno de mis servidores es usado por muchos usuarios, tengo una carga de CPU muy alta, y un tráfico de red muy alto, digamos que creo, por ejemplo que tengo la sospecha de que uno de los casos de falla es si uno de mis equipos tiene un trabajo que se atora en un bucle infinito. Entonces, si pienso que uno de los casos de falla es que una de mis máquinas, uno de mis servidores web (código de servidor) se atoró en un bucle infinito, por lo que aumenta la carga de CPU pero el tráfico de red no porque sólo lo está girando sus ruedas y haciendo mucho del trabajo del CPU, ya sabe, está atorado en un bucle infinito. En este caso, para detectar ese tipo de anomalía, puedo crear una nueva variable, x5 que puede ser la carga de CPU dividida entre el tráfico de red.



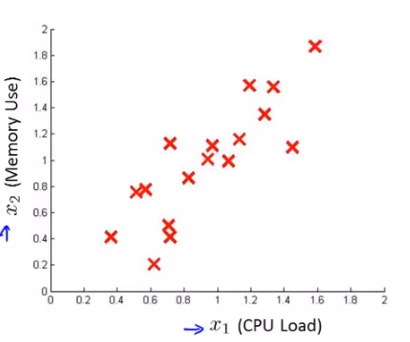
De ese modo, x5tomará un valor inusualmente grande, si una de las máquinas tiene una carga de CPU muy grande pero no tanto tráfico de red, esta será una variable que ayudará a su captura de detección de anomalías, un cierto tipo de anomalía. También puede ser creativo e inventar otras variables. Como tal vez tenga una función x6, que es la carga de CPU al cuadrado dividida entre el tráfico de red.



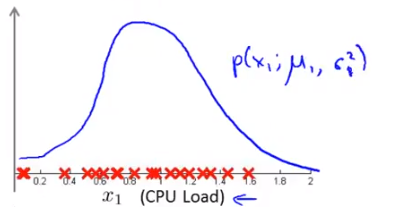
Esta sería otra variante de una variable, tal como x5 para tratar de capturar anomalías donde una de sus máquinas tiene una carga de CPU muy alta, que tal vez no tiene un tráfico de red proporcionalmente grande. Mediante la creación de funciones como estas, puede empezar a capturar anomalías que realmente corresponden a combinaciones inusuales de valores de las variables.   
  
En este video, hablamos de cómo crear y tomar una variable y tal vez transformarla un poco, de modo que se vuelva un poco más Gaussiana, antes de alimentarla en un algoritmo de detección de anomalías. También vimos el análisis del error en este proceso de creación de funciones para tratar de capturar diferentes tipos de anomalías. Espero que este tipo de pautas le ayuden a elegir buenas variables para añadir a su algoritmo de detección de anomalías y ayudarle así a capturar todo tipo de anomalías.

# Distribución gaussiana multivariante

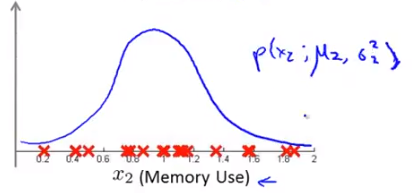
En este video y en el siguiente, me gustaría hablarles sobre una posible extensión del algoritmo de detección de anomalías que hemos desarrollado hasta ahora. Esta extensión utiliza algo llamado: distribución Gaussiana multivariante y tiene algunas ventajas y algunas desventajas, a veces puede captar algunas anomalías que el algoritmo anterior no pudo.   
  
Para motivar esto, vamos a empezar con un ejemplo, digamos que nuestros datos valores no asignados se parecen a lo que he trazado aquí:



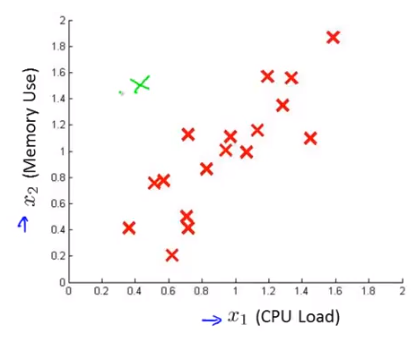
Y voy a usar el ejemplo de monitoreo de las máquinas en el centro de datos, la supervisión de equipos en el centro de datos. Así que mis dos variables son x1, que es la carga de CPU y x2, que es, tal vez, el uso de la memoria.   
  
Si tomo mis dos variables x1 y x2 y las modelo como Gaussianas, entonces aquí está un trazo de mis variables x1, aquí se muestra un trazo de mis variables x2, y si coloco una gaussiana en esto, quizá obtenga una una gaussina como esta, por lo que aquí está p(x1), que depende de los parámetros M1 y «sigma» cuadrada1:



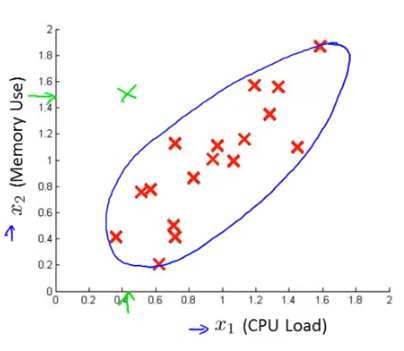
Y aquí está mi memoria usada y, como sabe, tal vez voy a conseguir una Gaussiana que se parezca a esto, y esta es mi p(x2) que depende de M2 y «sigma» cuadrada 2.



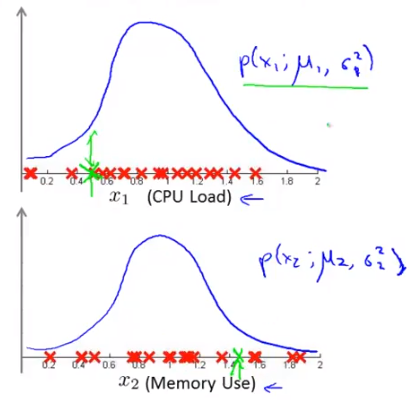
Entonces esto es algoritmo de detección de anomalías modela x1 y x2.   
  
Ahora, digamos que en los conjuntos de prueba tengo un ejemplo que se parece a esto.



La ubicación de esa cruz verde, en donde el valor de x1 es de aproximadamente 0.4 y el valor de x2 es de aproximadamente 1.5. Ahora, si observamos los datos, parece como, sí, la mayoría de los datos están en esta región:

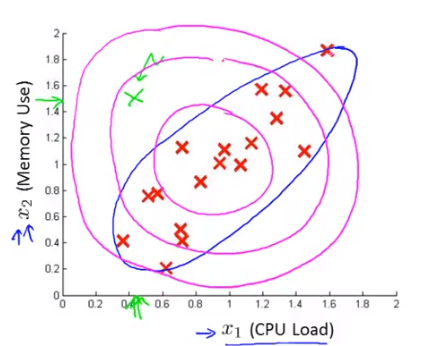


De modo que la cruz verde está bastante lejos de cualquiera de los datos que he visto. Parece que debe resaltarse como una anomalía. Así que en los los datos de mis buenos ejemplos, parece que, ya sabe, la carga del CPU y el uso de la memoria, de alguna manera crecen linealmente uno con el otro. Si tengo una máquina con una gran parte del CPU en uso, el uso de memoria también será alto, mientras que en este ejemplo, este ejemplo en verde aquí, la carga del CPU es muy baja pero el uso de memoria es muy alto, y no he visto eso antes en mi conjunto de entrenamiento. Parece que eso es una anomalía pero vamos a ver lo que el algoritmo de detección hará. Bueno, para la carga del CPU, se pone alrededor de 0.5 y está probabilidad razonablemente alta, no está tan lejos de otros ejemplos que hemos visto, tal vez, mientras que, para el uso de la memoria, esta asignación 0.5, para el uso de la memoria, es aproximadamente 1.5, que se muestra allí. Una vez más, como sabe, depende de nosotros, no es terriblemente Gaussiano pero este valor y este otro no son tan diferentes de otros ejemplos que hemos visto, entonces p(x1) será bastante alta, razonablemente alta y p(x2) también será razonablemente alta.

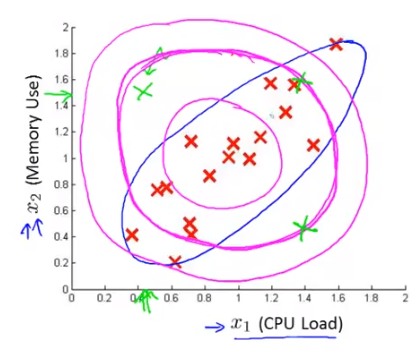


Quiero decir, he tenido ejemplos con aún más memoria utilizada o incluso con un menor uso de CPU y este ejemplo no parece tan anómalo.

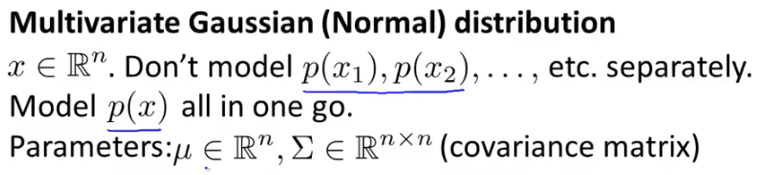
Y así, un algoritmo de detección de anomalías fallará en marcar este punto como una anomalía y resulta que lo que nuestro algoritmo de detección de anomalías está haciendo es que no se está dando cuenta de que este elipse azul muestra la región de alta probabilidad, es que, una de las cosas es que, los ejemplos aquí muestran una alta probabilidad y los ejemplos ejemplos del siguiente círculo, una menor probabilidad y estos ejemplos son de una probabilidad aún menor



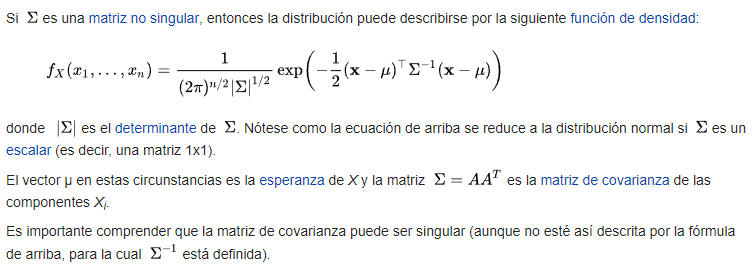
Y de alguna manera, aquí hay cosas que son, esa cruz verde, es una probabilidad bastante alta, y en particular, se tiende a pensar que, ya sabe, todo en esta región, todo lo que está del círculo del medio, tiene aproximadamente la misma probabilidad y no se da cuenta de que la probabilidad no es la misma para un punto que siga la linealidad de los ejemplos.



Así que para arreglar esto vamos a desarrollar una versión modificada del algoritmo de detección de anomalías usando algo llamado: distribución Gaussiana multivariante, también conocida como distribución normal multivariante.



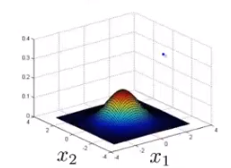
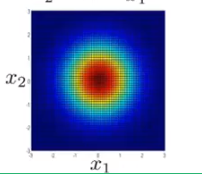
Así que esto es lo que vamos a hacer: tenemos variables "x" que están en Rn y en lugar de p(x1), p(x2), por separado, vamos a modelar p(x) de una sola vez, entonces modelamos p(x), ya sabe, todo al mismo tiempo.   
  
Los parámetros de la distribución Gaussiana multivariada son Mu, que es un vector, y «sigma», que es una matriz nxn, llamada una matriz de covarianza. Esto es muy similar a la matriz de covarianza que vimos cuando estábamos trabajando con el ACP, con el algoritmo de componentes principales.   
  
Para complementar esto, déjeme escribir la fórmula de la distribución Gaussiana multivariada, decimos que es la probabilidad de "x", y esto se parametriza por medio de parámetros: letra griega «Mu» y «sigma», por lo que la probabilidad de "x", es igual una vez más, no hay absolutamente ninguna necesidad de memorizar esta fórmula.   
Como sabe, puede revisarla siempre que necesite usarla pero así es como luce la probabilidad de "x":



El determinante de «sigma» es una función matemática de una matriz y realmente no necesita saber lo que es una determinante de una matriz, todo lo que necesita saber es que puede calcularlo en Octave, utilizando el comando "det" de «sigma». Bien y de nuevo, sólo para aclarar ¿de acuerdo? En esta expresión, estos «sigma»s aquí, son sólo una matriz nxn. Esto no es una suma y usted sabe, el «sigma» es una matriz nxn. Así que esa es la fórmula para p(x). Pero es aún más interesante, o más importante es cómo luce p(x) en realidad. Veamos algunos ejemplos de distribuciones Gaussianas multivariadas. Así que tomemos el mismo ejemplo dimensional, digamos que si tengo n=2, tengo dos variables, x1 y x2. Digamos que puse que «Mu» es igual a 0 y «sigma» es igual a esta matriz. Con 1s en las diagonales y 0s fuera de las diagonales, esta matriz a veces también se llama la matriz de identidad.

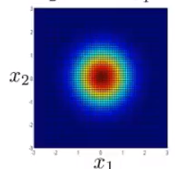
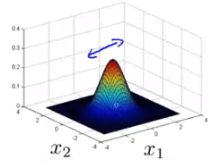


En ese caso, p(x) se verá así:

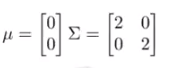
Y lo que les muestro en esta figura es, ya sabe, para un valor específico de x1 y para un valor específico de x2, la altura de esta superficie, el valor de p(x) Y con este ajuste de los parámetros p(x) es más alta cuando x1 y x2 son iguales a cero 0, ese es el pico de esta distribución de Gauss y la probabilidad disminuye con este tipo de Gaussiana bidimensional o esta superficie en forma de campana bidimensional cuando se aleja de su media [0 0].   
  
En el gráfico de la derecha aparece lo mismo pero trazado usando un trazo de contorno en lugar de usar diferentes colores y este rojo intenso a la mitad, corresponde a los valores más altos y a continuación, los valores disminuyen con el amarillo, siendo valores ligeramente menores, el cian muestra los valores más bajos y este azul profundo, los valores aún más bajos, por lo que esta es en realidad la misma figura, pero se traza vista desde la parte superior y con colores.   
  
Con esta distribución, puede ver que se enfrenta la mayor parte de la probabilidad cerca de 0,0 y luego, a medida que sale de 0,0, la probabilidad de x1 y x2 se reduce.   
  
Ahora vamos a tratar de variar algunos de los parámetros y veamos lo que sucede. Entonces, vamos a tomar «sigma» y lo cambiamos, así que digamos que «sigma» se reduce un poco.



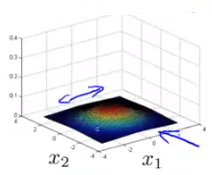


«sigma» es una matriz de covarianza, por tanto, mide la varianza o la variabilidad de las variables x1, x2. Si «sigma» se reduce, entonces lo que tiene, lo que se obtiene es que el ancho de esta protuberancia disminuye ya que los valores ahora se desvían menos de su media. La altura también se incrementa un poco, porque el área bajo la superficie es igual a 1. Entonces, la integral del volumen bajo la superficie es igual a 1, porque la distribución de la probabilidad se debe integrar a uno. Pero, si reduce la varianza, es un poco como la reducción de «sigma» al cuadrado, usted termina con una distribución más estrecha, y una que es un poco más alta. Y por lo que se ve en la figura de la derecha también las elipses concéntricas se han reducido un poco.

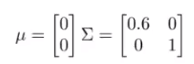
Por el contrario, si usted fuera a aumentar «sigma» a 2, 2 en las diagonales, esto es ahora dos veces la identidad,



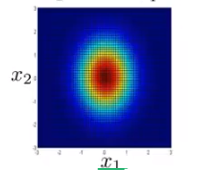
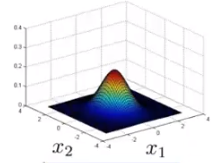
Entonces termina con una Gaussiana mucho más amplia y plana,



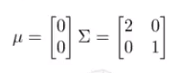
La anchura de esto es mucho más amplia. Esto es difícil de ver pero esta todavía es una protuberancia en forma de campana, sólo se ha aplanado mucho, se ha vuelto mucho más amplia y la varianza o la variabilidad de x1 y x2, sólo se hace más amplia.   
  
Aquí hay algunos ejemplos más: ahora vamos a tratar de variar uno de los elementos de «sigma» a la vez. Digamos que envío «sigma» a 0.6 allí, y 1 por allá.



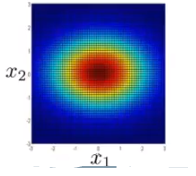
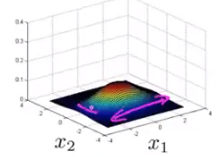
Lo que esto hace es reducir la varianza de la primera variable, x1 mientras mantiene la varianza de la segunda variable x2, la misma. Y así, con este ajuste de parámetros, puede modelar cosas de esta forma,



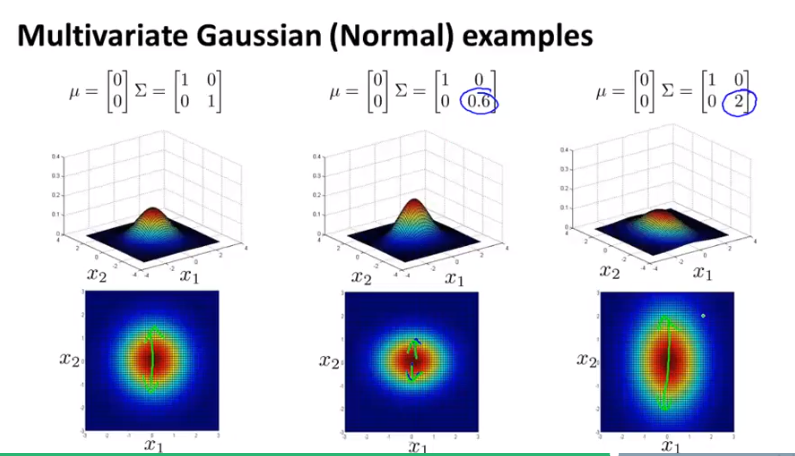
x1 tiene menor varianza, y x2 tiene mayor varianza. Mientras que si hago esto, si coloco esta matriz en 2, 1



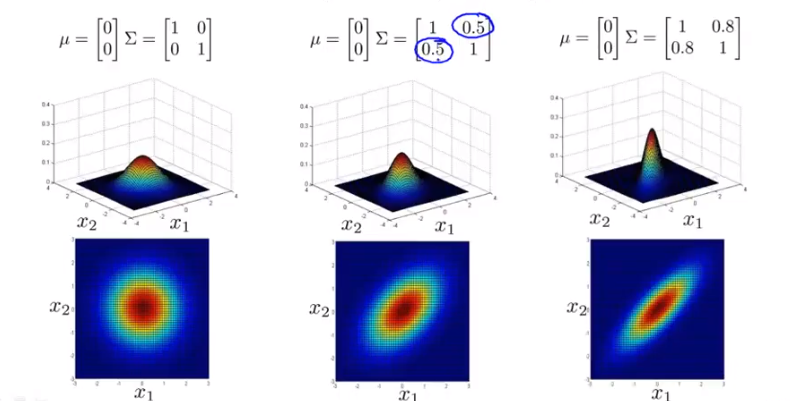
entonces usted también puede modelar ejemplos en los que, vamos a decir que x1 puede tener un gran rango de valores, mientras que x2 toma un rango relativamente estrecho de valores, Eso se refleja en esta figura también, ya sabe, donde la distribución cae más lentamente a medida que x1 se aleja de 0 y cae muy rápidamente a medida que x2 se aleja de 0.



De manera similar, si fuéramos a modificar este elemento de la matriz por el contrario, similar a la diapositiva anterior, excepto que aquí, donde se están desplazando, digamos que x2 puede tomar un rango muy pequeño de valores, entonces, si esto es 0.6, notamos ahora que x2 tiende a tomar un rango de valores menor que el ejemplo original, mientras que si tuviéramos que establecer «sigma» para ser igual a 2, es como decir que x2 tiene un alcance mucho más amplio de valores.

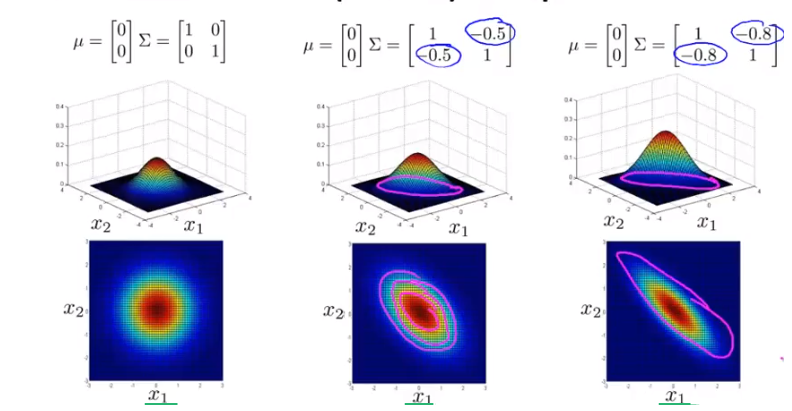


Ahora, uno de los mejores aspectos sobre la distribución Gaussiana multivariada es que también puede usarla para modelar correlaciones entre los datos. Es decir, podemos utilizarla para modelar el hecho de que x1 y x2 tienden a estar altamente correlacionados entre sí, por ejemplo. Específicamente, si comienza a cambiar las entradas fuera de la diagonal de esta matriz de covarianza, puede obtener un tipo diferente de distribución de Gauss. Y a medida que incremento las entradas fuera de la diagonal 0.5 a 0.8, lo que consigo es esta distribución que tiene un pico cada vez más delgado a lo largo de esta línea de "x=y".

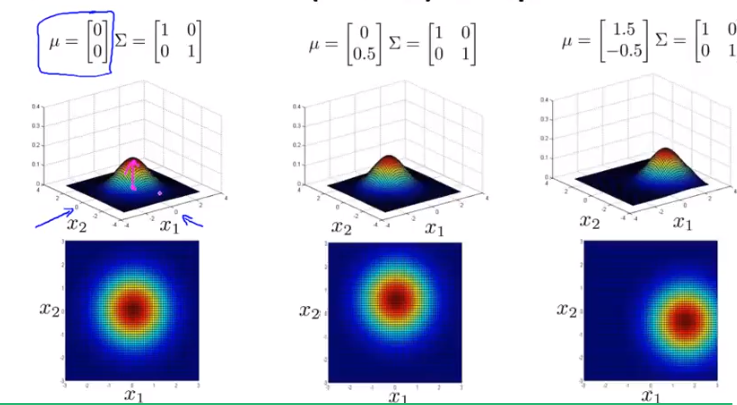


Entonces, aquí el contorno indica que "x" y "y" tienden a crecer juntas y las cosas que tienen una gran probabilidad son, ya sea que x1 sea grande y y2 sea grande o x1 sea pequeña y y2 sea pequeña. o algún punto intermedio y a medida que esta entrada 0.8 se hace grande, se obtiene una distribución de Gauss, que es como en lugar donde toda la probabilidad se encuentra en este tipo de región estrecha, donde "x" es aproximadamente igual a de "y". Esta es una distribución muy alta y delgada, la línea en su mayoría a lo largo de esta región central, donde "x" está cerca de "y".

Así que esta ponemos estas entradas como entradas positivas, en contraste, si colocamos estos en valores negativos, a medida que lo reduzco de - 0.5 hasta - 0.8, a continuación, lo que tenemos es un modelo en el que ponemos la mayor parte de la probabilidad en este tipo de región de correlación negativa x1 y x2 y así, la mayor parte de la probabilidad ahora se encuentra en esta región, donde x1 es aproximadamente igual a -x2, en lugar de x1 igual a x2. Y así, esto captura una especie de correlación negativa entre x1 y x2. Y entonces esto es espero que esto le de una idea de las diferentes distribuciones que la distribución Gaussiana multivariada puede capturar.



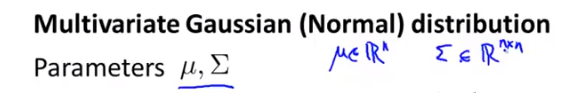
Al continuar con la variación de la matriz de covarianza «sigma», lo que también puede hacer es variar el parámetro Mu, y de forma operativa, tenemos «Mu» =0.0, por lo que la distribución se centró en torno a x1=0, x2=0, de modo que el pico de la distribución está aquí, mientras que, si variamos los valores de Mu, a continuación, esto varía el pico de la distribución y así, si «Mu» es igual a 0, 0.5, el pico está en, ya sabe, x1=0, y x2=0.5, de modo que el pico o el centro de esta distribución se ha cambiado y si «Mu» era 1.5 menos 0.5 entonces está bien, de manera similar, el pico de la distribución ahora se ha desplazado a una ubicación diferente, correspondiente al lugar donde, ya sabe, x1 es 1.5 y x2 es -0.5, y al variar el parámetro «Mu», sólo se mueve alrededor del centro de toda esta distribución.



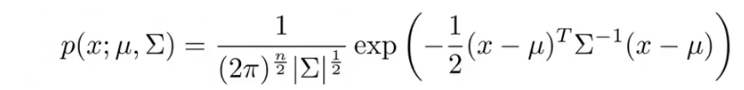
Espero que el mirar todas estas imágenes le de una impresión de los tipos de distribuciones de probabilidad que la distribución Gaussiana multivariada permite capturar y la principal ventaja de esto es que le permite capturar cuándo puede esperar que dos variables distintas se correlacionen de forma positiva o tal vez negativa. En el siguiente video, tomaremos esta distribución Gaussiana multivariada y la aplicaremos a la detección de anomalías.

# Detección de anomalías usando la distribución multivariada gausiana

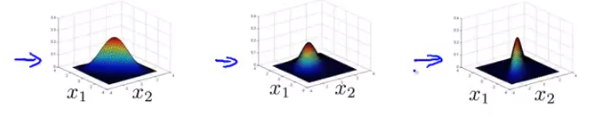
En el último vídeo, hablamos de la distribución Gaussiana multivariada, y vimos algunos ejemplos de los tipos de distribuciones que puede modelar, a medida que varía los parámetros, «Mu» y «sigma». En este video, tomaremos esas ideas y las aplicaremos para desarrollar un algoritmo de detección de anomalías.   
  
Para recapitular, la distribución Gaussiana multivariada o distribución normal multivariada, tiene dos parámetros, «Mu» y «sigma», donde «Mu» es un vector dimensional y «sigma»,   
la matriz de covarianza es una matriz nxn.



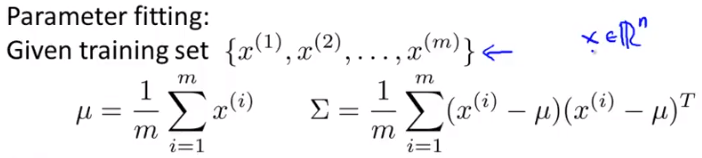
Esta es la fórmula para la probabilidad de "x"



Como se parametriza por medio de «Mu» y «sigma», a medida que varía «Mu» y «sigma», puede obtener un rango de diferentes distribuciones, como, ya sabe, estos son tres ejemplos de los que vimos en el video anterior:

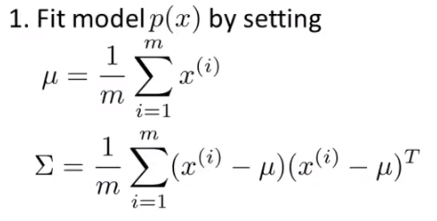


Vamos entonces a hablar del ajuste de parámetros o el problema de la estimación de parámetros. La cuestión, como siempre, es que si tengo conjunto de ejemplos de x1 a xm y aquí cada uno de estos ejemplos es un vector "n"dimensional y creo que mis ejemplos vienen de una distribución Gaussiana multivariada, ¿cómo puedo tratar de estimar los parámetros de «Mu» y «sigma»? Bueno, la fórmulas estándar para su estimación es que usted establece «Mu» para ser el promedio de sus ejemplos de entrenamiento y establece «sigma» para ser igual a esto, esto es en realidad como el «sigma» que habíamos escrito cuando estábamos utilizando el ACP o algoritmo de análisis de componentes principales.

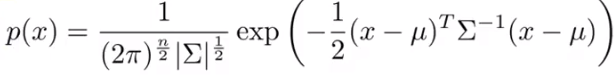


Entonces sólo tiene que activar estas dos fórmulas y esto le daría el parámetro estimado «Mu» y su parámetro estimado «sigma».   
  
Así que dado un grupo de datos, así es cómo usted estima «Mu» y «sigma». Tomemos este método y apliquemoslo a un algoritmo de detección de anomalías. Entonces, ¿cómo reunimos todo esto para desarrollar un algoritmo de detección de anomalías? Esto es lo que hacemos:

Primero tomamos nuestro conjunto de entrenamiento y ajustamos el modelo, ajustamos p(x), al establecer «Mu» y «sigma», como se describió en la diapositiva anterior.



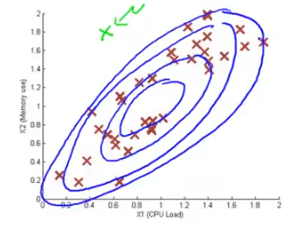
Luego, cuando se le da un nuevo ejemplo x, si si se le da un ejemplo de prueba, vamos a tomar un ejemplo anterior para tener un nuevo ejemplo aquí. Ese es mi ejemplo de prueba. Dado el nuevo ejemplo x, lo que haremos será calcular p(x), usando esta fórmula para la distribución Gaussiana multivariada:



Y entonces, si p(x) es muy pequeña, entonces marcamos esto como una anomalía, mientras que, si p(x) es mayor que el parámetro «épsilon», entonces no lo marcamos como anomalía.

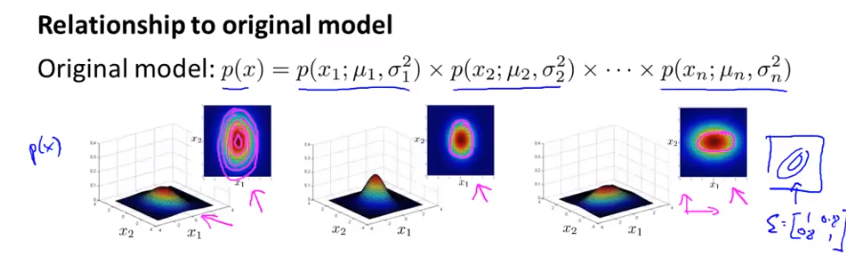


Así resulta que, si tuviéramos que ajustar una distribución Gaussiana multivarianda para este conjunto de datos, sólo las cruces rojas, no las verdes de ejemplo, termina con una distribución gaussiana que coloca mucha probabilidad en la región central, ligeramente menos probabilidad aquí, ligeramente menos probabilidad aquí, ligeramente menos probabilidad aquí, y muy baja probabilidad en el punto que está afuera.

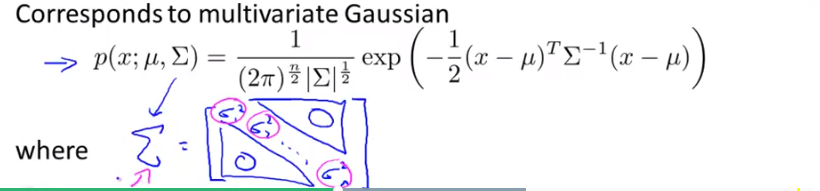


Y si aplica la distribución Gaussiana multivariada a este ejemplo, en realidad marcará correctamente ese ejemplo como una como una anomalía.   
  
Por último, vale la pena decir unas pocas palabras acerca de lo que es la relación entre el modelo de distribución Gaussiana multivariada y el modelo original, en el que modelamos p(x) como producto de esta p(x1), p(x2), hasta p(xn). Resulta que usted puede demostrar matemáticamente, no voy a hacer la prueba aquí pero usted puede demostrar matemáticamente esta relación entre el modelo de distribución Gaussiana multivariada y este original. En particular, resulta que el modelo original corresponde a Gaussianas multivariadas, en las que los contornos de las mismas siempre están alineados al eje.

Así que estos son tres ejemplos de distribuciones Gaussianas que puede ajustar utilizando el modelo original. Resulta que esto corresponde a Gaussianas multivariadas, en las que como sabe, la elipse aquí, los contornos de esta distribución, tenemos que este modelo realmente corresponde a un caso especial de distribución Gaussiana multivariada y en particular, este caso especial se define al restringir la distribución de p(x), la distribución Gaussiana multivariada de p(x), de modo que que los contornos de la función de densidad de probabilidad, de la función de distribución de probabilidad, están alineados los ejes. Y para que pueda obtener una p(x) con una Gaussiana multivariada que se ve así o como los otros dos ejemplos- Puede darse cuenta que en los 3 ejemplos, estas elipses, o estos óvalos que estoy dibujando, tienen sus ejes alineados con los ejes x1, x2 y lo que no tenemos, es un conjunto de contornos que están en un ángulo, ¿verdad? (las covarianzas o lo que está fuera de la diagonal principal en la matriz de varianzas covarianzas es igual a 0) Y esto corresponde a ejemplos en los que «sigma» es igual a 1 1, 0.8, 0.8, digamos que con elementos distintos a 0 fuera de las diagonales.

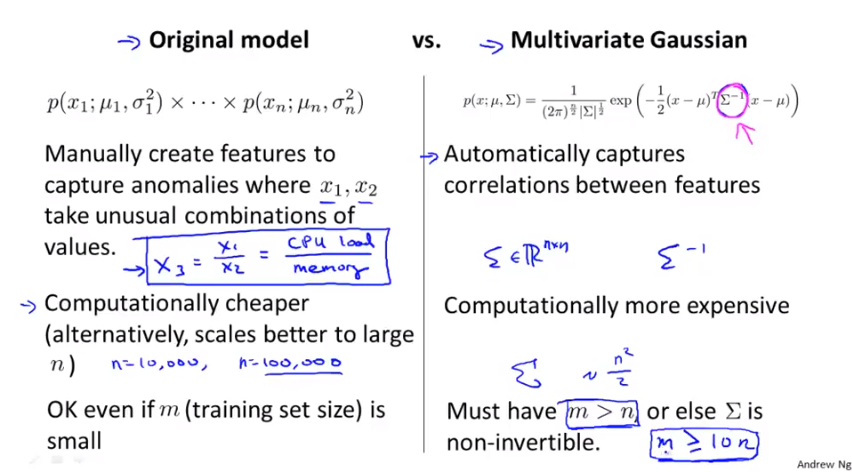


También resulta que, es posible demostrar matemáticamente que este modelo es en realidad la misma distribución Gaussiana multivariada pero con una restricción. La restricción es que la matriz de covarianza «sigma» debe tener ceros en los elementos fuera de la diagonal. En particular, la matriz de covarianza de «sigma», esto de aquí, podría ser «sigma» cuadrada 1,«sigma» cuadrada 2, hasta «sigma» al cuadrado n y luego todo en las entradas fuera de la diagonal, todos estos elementos por encima y por debajo de la diagonal de la matriz, todos estos van a ser cero. De hecho, si toma estos valores de «sigma», «sigma» cuadrada 1, «sigma» cuadrada 2 hasta «sigma» cuadrada n, y los conecta aquí, ya sabe, conectarlos a esta matriz de covarianza, entonces los dos modelos son en realidad idénticos.



Es decir, este nuevo modelo, que usa la distribución Gaussiana multivariada corresponde exactamente al modelo antiguo, si la matriz de covarianza «sigma» sólo tiene elementos 0 fuera de las diagonales y en las imágenes esto corresponde a tener distribuciones Gaussianas, donde los contornos de esta función de distribución de probabilidad, están alineados a los ejes. Así que no se le permite modelar las correlaciones entre las distintas variables.   
  
Así que en ese sentido, el modelo original en realidad es un caso especial de este modelo Gaussiano multivariante. Así que ¿cuando utilizaría cada uno de estos dos modelos? ¿cuándo se usa el modelo original y cuándo se usa el modelo Gaussiano multivariado?   
  
El modelo original se usa probablemente un poco más a menudo, mientras que la distribución Gaussiana multivariada se utiliza un poco menos pero tiene la ventaja de ser capaz de capturar las correlaciones entre las variables. De modo que si quiere capturar anomalías donde tiene diferentes variables, digamos que en donde las variables x1, x2 toman combinaciones poco usuales de valores, por lo que en ejemplo anterior, tuvimos ese caso donde la anomalía fue con la carga de CPU y el uso de memoria teniendo combinaciones poco usuales de valores, si desea utilizar el modelo original para captar eso, entonces lo que necesitamos hacer es crear una variable extra, como x3=x1/x2 ya sabe, igualarlo a quizá, la carga de CPU dividida entre la memoria utilizada, o algo así, y necesitará crear variables adicionales si hay combinaciones poco usuales de valores, donde x1 y x2 toman una combinación inusual de valores a pesar de que x1 por sí misma y x2 por sí misma, lucen como si tomaran un valor perfectamente normal pero si usted está dispuesto a invertir el tiempo para crear de forma manual una variable adicional de este tipo, entonces, el modelo original trabajará bien.

Mientras que por el contrario, el modelo Gaussiano multivariante puede capturar de forma automática las correlaciones entre diferentes variables pero el modelo original tiene algunas otras ventajas más significativas también. Una enorme ventaja del modelo original es que es computacionalmente más barato, otra forma de comprender esto, es que escala mejor a valores muy grandes de "n" y números muy grandes de variables y aún si "n" fuera 10,000, o incluso si "n" fuera igual a 100,000, el modelo original a menudo funcionará bien. Mientras que por el contrario con el modelo Gaussiano multivariante, note aquí, por ejemplo, que necesitamos calcular la inversa de la matriz «sigma», donde «sigma» es una matriz nxn, al calcular «sigma», si «sigma» es una matriz de 100,000x100,000 va a ser muy costoso computacionalmente. Y así, el modelo Gaussiano multivariado escala con menor eficiencia a grandes valores de "n" y finalmente, para el modelo original resulta que funciona bien, incluso si tiene un conjunto de entrenamiento relativamente pequeño, que son estos pequeños ejemplos sin valores asignados que utilizamos para modelar p(x), por supuesto, y esto funciona bien, incluso si «Mu» es, ya sabe, tal vez 50, 100, funcionará bien. Mientras que la distribución Gaussiana multivariada es una clase de propiedad matemática del algoritmo que debe tener m mayor que n, de modo que el número de ejemplos es mayor que el número de variables que tiene. Y hay una propiedad matemática de la forma en que calculamos los parámetros, y si esto no es cierto, si m es menor o igual a n, entonces esta matriz no es invertible, incluso, esta matriz es singular, por lo que ni siquiera se puede utilizar el modelo Gaussiano multivariado, a menos que haga algunos cambios en él. Pero una típica regla general que yo uso es que voy a utilizar el modelo Gaussiano multivariado sólo si m es mucho mayor que n, por lo que esta es una especie de requisito matemático estrecho, sin embargo, en la práctica, usaría el modelo Gaussiano multivariado, sólo si m fuera un poco mayor que n. Si m fuera mayor o igual a 10 veces n, digamos que, podría ser una regla razonable general, y si no satisface esto, entonces el modelo Gaussiano multivariante tiene muchos de los parámetros, ¿no?, por lo que la matriz de covarianza «sigma» es una matriz nxn, por lo que tiene, más o menos parámetros al cuadrado n, porque es una matriz simétrica, en realidad está más cerca de n al cuadrado sobre 2 parámetros pero estos son muchos parámetros, por lo que necesita asegúrarse de que tiene un volumen suficientemente grande para m, asegúrese de que tiene los datos suficientes para ajustar todos estos parámetros. M mayor que 0 o igual a 10n sería una regla razonable general para asegurarse de que puede estimar esta matriz de covarianza «sigma» de forma razonable.



Así que en la práctica, el modelo original mostrado a la izquierda es el que se usa con más frecuencia. Si usted sospecha que es necesario capturar las correlaciones entre las variables, lo que la gente suele hacer es diseñar manualmente funciones adicionales como x3 para capturar combinaciones específicas poco usuales de valores, no obstante,en problemas en los que tiene un conjunto de entrenamiento muy grande o m es muy grande y "n" no es demasiado grande, entonces vale la pena considerar el modelo Gaussiano multivariante y puede funcionar mejor también además de salvarle de tener que pasar el tiempo creando manualmente variables adicionales en caso de que las anomalías resulten ser capturadas por combinaciones de valores inusuales de las variables.   
  
Finalmente, sólo quiero mencionar brevemente una propiedad algo técnica pero si usted está ajustando el modelo Gaussiano multivariado y si se da cuenta de que la matriz de covarianza «sigma» es singular, o si descubre que no es invertible, por lo general hay 2 casos para esto:

Uno es que si está fallando al satisfacer esta condición de m mayor que n

El segundo caso es si tiene variables redundantes. Por variables redundantes, me refiero a que si tiene 2 variables que son las mismas, de alguna manera, por accidente ha realizado dos copias de la función, por lo que su x1 es exactamente igual a x2 o si tiene variables redundantes, como por ejemplo, su variable x3 es igual a x4 más la variable x5. ¿De acuerdo?, así que si tiene variables altamente redundantes como estas, donde si x3 es igual a x4 + x5, bueno, x3 no contiene ninguna información extra, ¿verdad? así que sólo tome estas otras 2 variables y agréguelas juntas y si tienes este tipo de variables redundantes, variables duplicadas o esta clase de variables, entonces, «sigma» puede ser no invertible.   
  
Existe un conjunto de depuración, esto debe ocurrir con poca frecuencia, por lo que probablemente no se encontrará con esto, es muy poco probable que usted tenga que preocuparse por esto pero en caso de implementar un modelo Gaussiano multivariado, encontrará que «sigma» no es invertible. Lo que quiero hacer es primero asegúrarme de que m es un poco más grande que n, y si es así, lo segundo que haré es simplemente revisar las variables redundantes. Si hay 2 variables que son iguales, sólo las voy a eliminar o si tiene variables redundantes, si x3 es igual a x4 más x5, simplemente hay que deshacerse de la variable redundante y entonces debería funcionar bien de nuevo.

Como un apartado para aquellos de ustedes que son expertos en el álgebra lineal, por variables redundantes, me refiero al término formal de que son variables linealmente dependientes. Sin embargo, en la práctica, lo que realmente significa es uno de esos problemas de saturación del algoritmo, si sólo crea variables no redundantes, eso resolvería el problema de que «sigma» no es invertible. Una vez más, las probabilidades de que se encuentre con esto son muy bajas, así que quizá puede sólo aplicar el modelo de distribución Gaussiana multivariada sin tener que preocuparse porque «sigma» no sea invertible, siempre y cuando m sea mayor o igual a n. De modo que detección de anomalías con la distribución Gaussiana multivariada, si usted aplica este método sería capaz de tener un algoritmo de detección de anomalías, que de forma automática captura las correlaciones negativas y positivas entre sus diferentes variables y marca una anomalía si detecta una combinación inusual de los valores de las variables.