Machine Learning No Supervisado

Gaussian Mixtures - Kernel Density Estimation

Tomás Fontecilla

7 de octubre de 2022

Mezclas Gaussianas

Muchas veces ocurre que podemos modelar datos respecto de una distribución gaussiana. Las ventas, riesgo de contraer una enfermedad, ingresos, altura, unidades vendidas son algunos ejemplos.

Por tanto es muy natural asumir que los clusters que uno pueda detectar vengan de diferentes distribuciones gaussianas, o en otras palabras, intentamos modelar los datos *como si fueran* distribuciones gaussianos.

En términos de una dimensión, la densidad de probabilidad de una función de distribución **gaussiana** está dada por:

Función de densidad distribución de Gauss

$$f(X|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

donde μ y σ^2 son la media y la varianza respectivamente

Pero esta densidad sólo representa una gaussiana. Extendiendo la función anterior, obtendríamos, en formato matricial:

Función de densidad distribución de Gauss multivariada

$$f(X|\mu,\sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)}$$

donde μ es el vector d dimensional de medias y Σ es d X d matriz de covarianza.

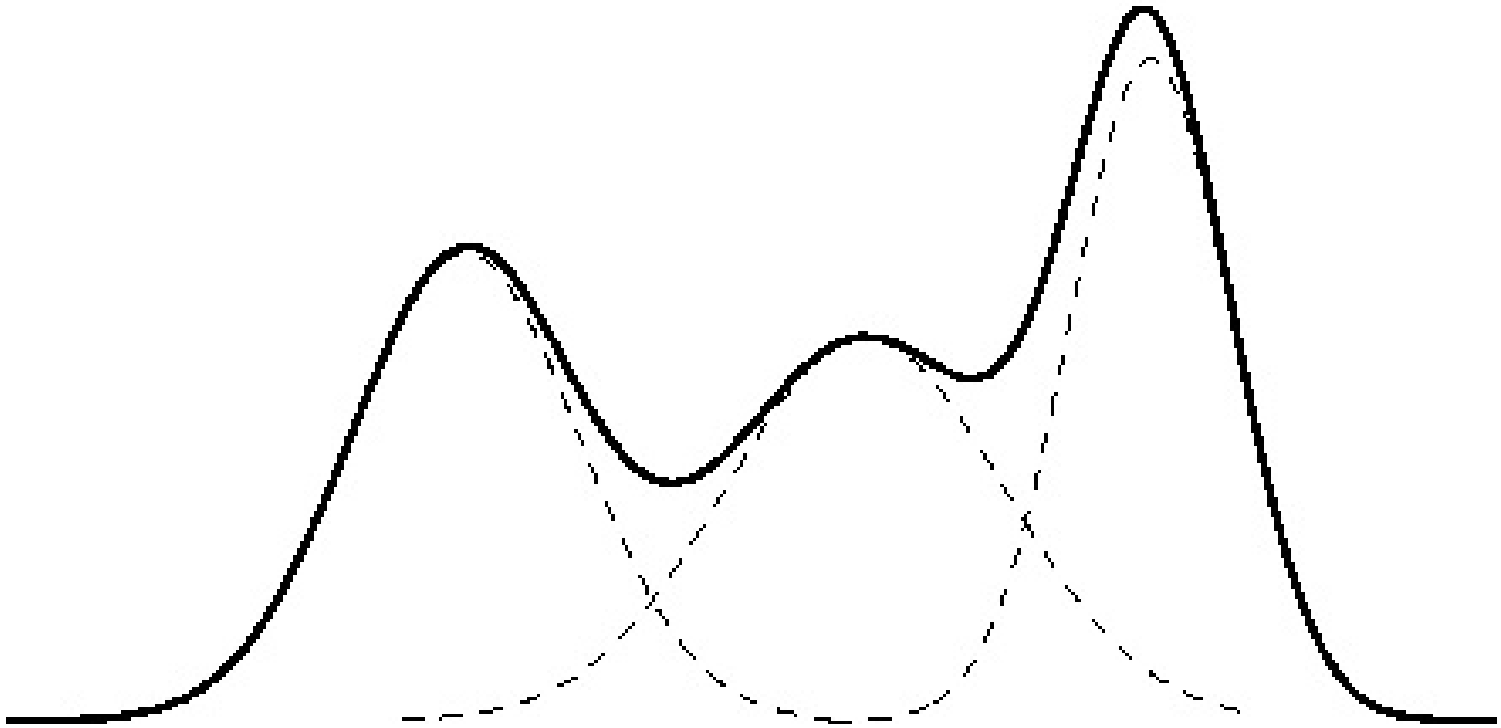
Mezclas Gaussianas

Supongamos entonces que tenemos K clusters (supondremos que sabemos K). Por tanto, μ y Σ se pueden estimar para cada k. Normalmente usariamos **método de máxima verosimilitud** para estimar μ y σ^2 si fuera unidimensional, pero como tenemos k clusters y la densidad de probabilidad está definida como una función lineal de densidades para todas las K distribuciones, donde π_k es el coeficiente para la k-ésima distribución

$$p(X) = \sum_{k=1}^{K} \pi_k f(X|\mu_k, \Sigma_k)$$

Mezclas de gaussianas - gráfico

El siguiente gráfico muestra cómo se ve la densidad de una distribución de mezcla de gaussianas.



Como podemos ver, es algo que ya conocen!

Mezclas Gaussianas - Estimadores máximo verosímil

Para estimar los parámetros por método de máxima verosimilitud, hay que calcular $L(X|\mu, \Sigma, \pi) = \Pi \mathbb{P}(X_i|\mu, \Sigma, \pi)$
De la forma tradicional, tenemos que:

$$lnL(X|\mu, \Sigma, \pi) \begin{cases} = \sum_{i=1}^{N} \mathbb{P}(X_i) \\ = \sum_{i=1}^{N} ln \sum_{k=1}^{K} \pi_k f(X_i|\mu_k, \Sigma_k) \end{cases}$$

definimos una variable aleatoria $\gamma_k(X) = \mathbb{P}(k|X)$. Por el teorema de Bayes:

$$\gamma_k(X) = \frac{\mathbb{P}(X|k)\mathbb{P}(k)}{\sum_{k=1}^{K} \mathbb{P}(k)\mathbb{P}(X|K)} = \frac{\mathbb{P}(X|k)\pi_k}{\sum_{k=1}^{K} \pi_k \mathbb{P}(X|K)}$$

Mezclas Gaussianas - Estimadores máximo verosímil

Ahora, para obtener el EMV, las derivadas respecto de μ, Σ y π e igualamos a cero.

Estimador Máximo Verosímil

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n) x_n}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n)(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_k(x_n)$$

Donde $\sum_{n=1}^{N} \gamma_k(x_n)$ es el total de muestras en el k-ésimo grupo y N es el número total de muestras.

Mezcla de Gaussianas - Algoritmo EM

Tenemos un problema. Ninguno de los cálculos previos es de forma cerrada, por lo que te tendremos que aplicar un algoritmo para poder determinar sus valores.
Algoritmo iterativo para encontrar las estimaciones máximo verosímiles de los parámetros del modelo cuando los datos están incompletos o existen datos faltantes o tiene variables ocultas.
EM escoge valores al azar para datos faltantes y estima un nuevo set de datos. Esos nuevos valores son recursivamente utilizados para estimar nuevos valores hasta llenar todos los faltantes y que ellos se vuelvan fijos.

Algoritmo EM

Existen dos pasos en el algoritmo: Paso Estimación (o Esperanza) y Paso Maximización

Algoritmo EM

- Paso Estimación:
  - inicializa μ_k, Σ_k y π_k con valores aleatorios o como resultados de kmeans o clustering jerárquico.
  - Para esos parámetros, estima valores de las variables latentes (γ_k)
- Paso Maximización:
  - Actualiza los valores de los parámetros calculados usando máxima verosimilitud.
  - "rinse & repeat"

Esto se hará hasta convergencia.

Mezclas de Gaussianas

¿ Cuál es su principal ventaja?

- Tiene una distribución de probabilidad en la base del modelo
- utiliza un algoritmo especializado para encontrar los parámetros
- ajusta áreas referentes a los datos, no a figuras externas a ellos. entonces no necesariamente hace cluster sobre esferas.
- Tiene funciones de probabilidad en su base

¿ Principales Desventajas?

- Requerie definir el número de clusters
- El algoritmo EM depende de su inicialización.

Estimación de densidad por Núcleo

Kernel Density Estimation o Estimación de densidad de Núcleo es otra medida de cómo podemos estimar la función de densidad de probabilidad de una variable aleatoria, basado en kernels como ponderadores.

Es una técnica de suavizamiento donde se realiza inferencia respecto de la población basado en una muestra finita de datos.

Una aplicación usual es la estimación de densidades marginales condicionales por clase de datos cuando usamos Naive Bayes, mejorando la certeza de la predicción.

Estimación de densidad por Núcleo - Definición

Sea $(x_1, ..., x_n)$ muestras aleatorias independiente e identicamente distribuidas extraidas de una distribución univariada con una densidad desconocida $\boldsymbol{f}$ para cualquier x. Nos interesa *estimar* la forma que tiene esta función.

KDE: Kernel density estimator

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

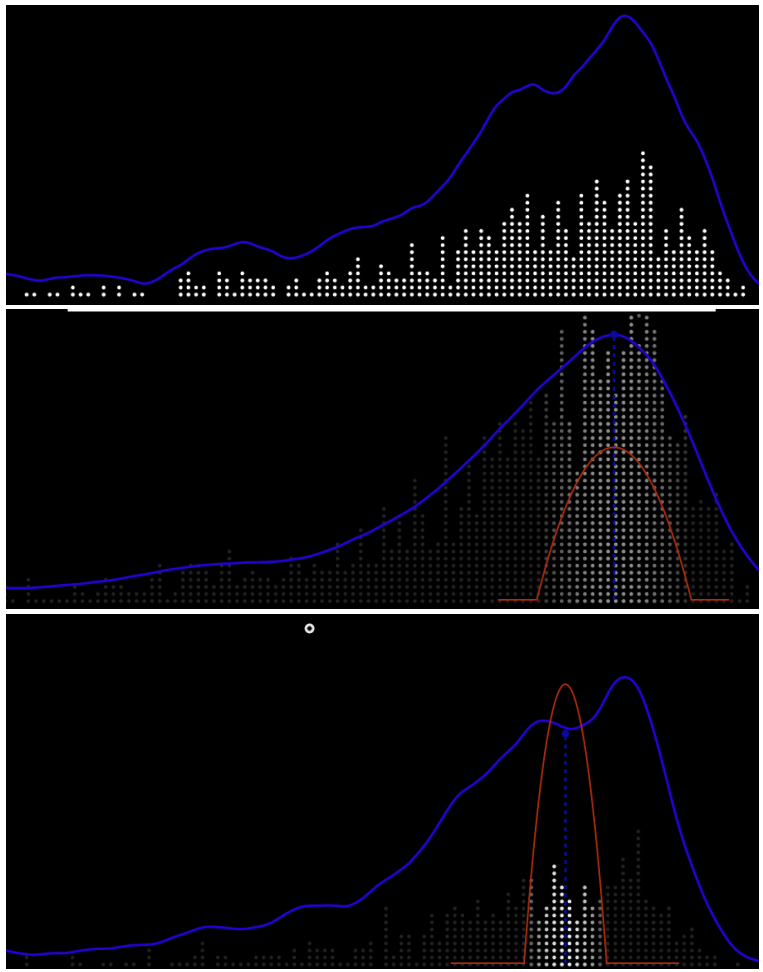
donde K es el kernel y $h > 0$ es un parámetro de suavizamiento llamado ancho de banda.
Un kernel con subíndice h es llamado kernel escalado y se define como $K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$

Estimación de densidad por Núcleo

En la práctica,

- supongamos que los puntos blancos son observaciones obtenidas de una distribución.
- A medida que los puntos se van acumulando, la silueta de ellos se parecer a a una distribución, pero no podemos saber su valor real.
- la línea azul muestra una estimación de la distribución subyacente, producido por la KDE.
- El parámetro *bandwidth* afecta que tan "suave" es la curva resultante.
- La KDE es calculada ponderando las distancias de todos los puntos que se han visto para cada lugar en la línea azul.

- la bandwidth afecta la distribución roja en la segunda figura.



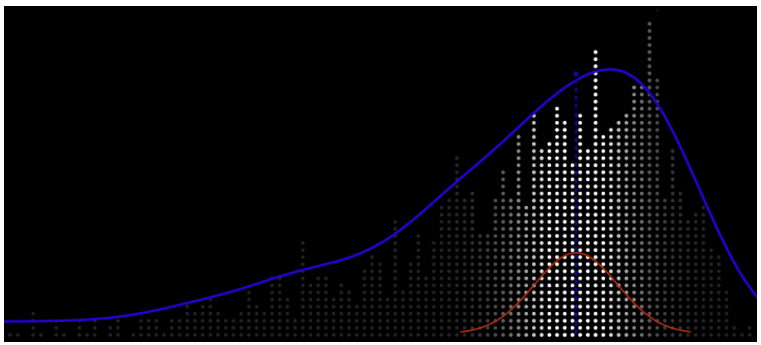
Estimación de densidad por Núcleo

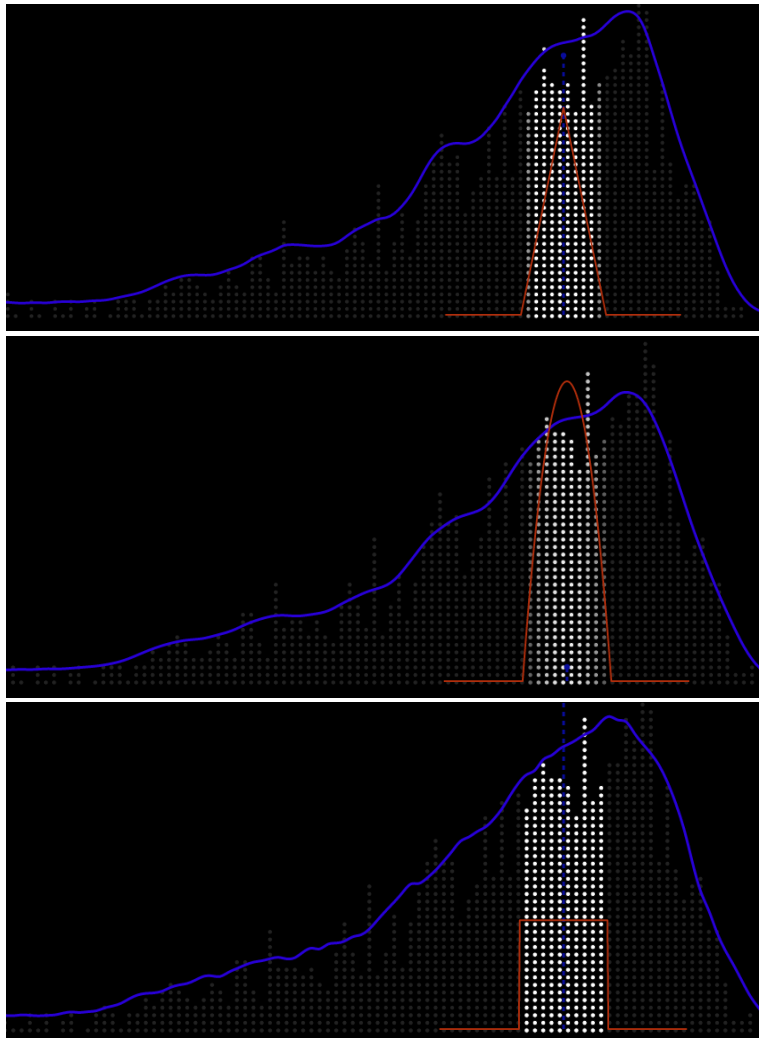
¿Como determinamos las ponderaciones? revisitemos la función de definición, pero cambiemos los nombres:

KDE

$$\hat{f}(x) = \sum_{observaciones} K\left(\frac{x - \text{observación}}{\text{bandwith}}\right)$$

K representa la función kernel. Distintos kernel producen diferentes estimaciones.





KDE - Usos

Como vimos, KDE busca la función de densidad de una distribución. Así algunos de sus usos son:

- Podemos hacer que Naïve Bayes no sea tan Naïve al permitir que las variables tengan una densidad conjunta por clase y no sean independientes.
- Podemos continuizar una variable discreta, permitiendo visualizar datos en gráficas de modos distintos.