Aprendizaje No Supervisado

Maestría en Ciencia de Datos

Lucas Fernández Piana

Primavera 2022

Universidad de San Andrés

T-SNE

INTRO

t-sne significa t distributed Stochastic Embedding.

INTRO

t-sne significa **t distributed Stochastic Embedding**.

- Es una técnica de reducción de dimensión que es utilizada sólo para visualización.
- Al transformar los datos trata de preservar la estructura local de los datos y a la vez tratar de preservar la estructura global lo más posible. Aunque este último objetivo es el que más se sacrifica.

KULLBACK-LEIBER

Es una forma de medir similaridad entre distribuciones

$$D_{KL}(p||q) = \sum_i p(i) log\left(\frac{p(i)}{q(i)}\right), \text{ distribuciones discretas.}$$

$$D_{KL}(p||q) = \int p(x) log\left(\frac{p(x)}{q(x)}\right), \text{ distribuciones continuas.}$$

KULLBACK-LEIBER

Cumple algunas propiedades:

- $D_{KL}(p||q) \geq 0$.
- $D_{KL}(p||q) = 0$ si $p = q$ c.t.p. ($p = q$ en el caso discreto).
- No es simétrica. Aunque puede simetrizarse facilmente como

$$D_{KL}(p, q) = \frac{1}{2} \left(D_{KL}(p||q) + D_{KL}(q||p) \right).$$

¿CÓMO FUNCIONA?

Primero veamos el primer algoritmo que surgió SNE, así vamos a entender mejor el papel de la t-student en esta historia.

¿CÓMO FUNCIONA?

Primero veamos el primer algoritmo que surgió SNE, así vamos a entender mejor el papel de la t-student en esta historia.

El objetivo es tratar de empatar la distribución de los datos en el espacio de alta dimensión con la distribución de la representación en baja dimensión.

¿CÓMO FUNCIONA?

Primero veamos el primer algoritmo que surgió SNE, así vamos a entender mejor el papel de la t-student en esta historia.

El objetivo es tratar de empatar la distribución de los datos en el espacio de alta dimensión con la distribución de la representación en baja dimensión.

Es decir,

Sea $D = \{x_1, \ldots, x_n\}$ mi conjunto de datos de dimensión N (grande).

¿CÓMO FUNCIONA?

Primero veamos el primer algoritmo que surgió SNE, así vamos a entender mejor el papel de la t-student en esta historia.

El objetivo es tratar de empatar la distribución de los datos en el espacio de alta dimensión con la distribución de la representación en baja dimensión.

Es decir,

Sea $D = \{x_1, \ldots, x_n\}$ mi conjunto de datos de dimensión N (grande).

Queremos construir una representación de los datos $D = \{y_1, \ldots, y_n\}$ con dimensión $q << N$.

SNE

Supongamos que ambos conjuntos de datos tienen distribución Gausiana, construímos las probabilidades condicionales:

$$p_{j|i} = \frac{\exp(-1/2||x_i - x_j||^2/\sigma_i^2)}{\sum_{l \neq i} \exp(-1/2||x_l - x_i||^2/\sigma_i^2)}.$$

$$q_{j|i} = \frac{\exp(-1/2||y_i - y_j||^2)}{\sum_{l \neq i} \exp(-1/2||y_l - y_i||^2}.$$

SNE

Finalmente, pedimos que los y's que estamos costruyendo en el espacio de menor dimensión minimizen la suma de las divergencias Kullback-Leibel:

$$C(Y) = C(y_1, \ldots, y_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{j|i} log \left(\frac{p_{j|i}}{q_{j|i}} \right).$$

SNE

Dada la función de costo, C, SNE usa la técnica de gradiente descendiente para minimizarla,

$$Y^{t+1} = Y^t - \eta \frac{\partial C}{\partial Y}(Y^t) + \alpha(t)(Y^{t-1} - Y^{t-2}), \text{ donde}$$

- $\alpha(t)$ se lo llama *Momento* de la iteracción t.
- η *learning rate*.
- $\frac{\partial C}{\partial y_i}(Y) = 2\sum_j (p_{i|j} - q_{i|j} + p_{j|i} - q_{j|i})(y_i - y_j)$.

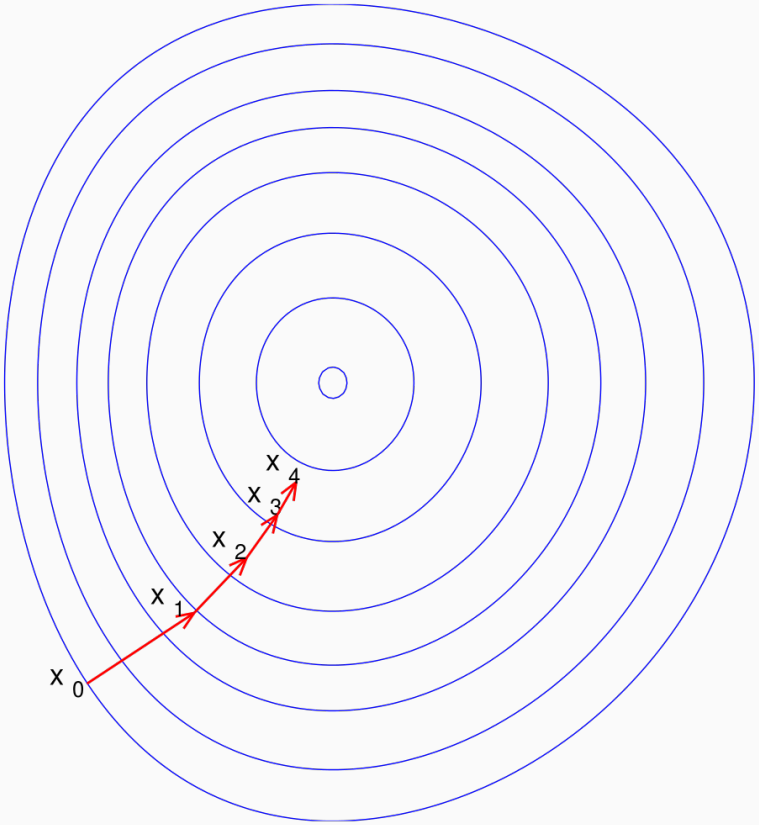

Figura 1: Esquema gradiente descendente

SNE

SNE tiene dos problemas difíciles de resolver:

- La función de costos es difícil de optimizar, gradiente descenden falla.
- ”Crowding Problem”: no hay suficiente espacio en dimensión baja para acomodar los vecinos de un dato que propone SNE. Es un problema común con otros algoritmos también.

t-SNE

El algoritmo t-SNE incorpora algunas modificaciones para lidiar con este lío de la idea original.

- Primero usar una versión simetrizada de las probabilidades $p_{ij} = \frac{1}{2N}(p_{i|j} + p_{j|i})$ y lo mismo para las q_{ij}.
- Reemplazar la distribución del espacio de dimensión baja por una t-student, $p(y) = \frac{1}{1+y^2}$. EL objetivo es tener menos problema de hacinamiento (crowding problem) al hacer más pesada la cola de la distribución.

t-SNE

Se redefinen las q_{ij} como

$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{l \neq i} \left(1 + ||y_l - y_i||^2\right)^{-1}}.$$

t-SNE

Se redefinen las q_{ij} como

$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{l \neq i} \left(1 + ||y_l - y_i||^2\right)^{-1}}.$$

El nuevo gradiente de la función de costos es,

$$\frac{\partial C}{\partial y_i}(Y) = 4 \sum_j (p_{ij} - q_{ij}) \frac{(y_i - y_j)}{1 + ||y_i - y_j||^2}.$$

t-SNE

Se redefinen las q_{ij} como

$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)^{-1}}{\sum_{l \neq i} \left(1 + ||y_l - y_i||^2\right)^{-1}}.$$

El nuevo gradiente de la función de costos es,

$$\frac{\partial C}{\partial y_i}(Y) = 4 \sum_j (p_{ij} - q_{ij}) \frac{(y_i - y_j)}{1 + ||y_i - y_j||^2}.$$

Más info intuítica de como usarlo,
https://distill.pub/2016/misread-tsne/.

t-sne y uma, a diferencia de PCA, si viene un dato nuevo, no puedo transformarlo, debo entrenar de nuevo el algoritmo para todos los datos.

COFFEE BREAK!