Universidad de Oviedo

Máster en Análisis de Datos e Inteligencia de Negocios

Trabajo Fin de Máster

Visualización de conjuntos de datos de alta dimensionalidad con t-SNE

Cristian Robledo Lete

Oviedo, Julio de 2019

Visualización de conjuntos de datos de alta dimensionalidad con t-SNE

Resumen

t-Distributed Stochastic Neighbor Embedding (t-SNE) es una técnica no lineal de análisis no supervisado de reducción de la dimensión muy utilizada a la hora de realizar visualizaciones en dos dimensiones acerca de conjuntos de datos de alta dimensionalidad. A pesar de tratarse de una técnica relativamente reciente, no existe mucha documentación acerca de ella y sus aplicaciones. No obstante, una de las más apreciadas por sus usuarios es su gran utilidad a la hora de buscar estructuras o agrupaciones dentro de un conjunto de datos, ya que representa muy bien aquellos grupos bien definidos y separados en el plano. Sin embargo, se trata de un método muy dependiente del tipo de individuos u objetos que se buscan representar, lo que requiere también tener un conocimiento previo de los datos, alejándose así del propio análisis no supervisado.

Visualization of high-dimensional datasets with t-SNE

Abstract

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear unsupervised dimensionality reduction technique used for make high-dimensional datasets' visualizations in two dimensions. In spite of being a relatively recent technique, it does not exist much documentation about it and its applications. However, one of the most appreciated by its users is its big utility when it is come to find structures and groups in a dataset, since it represents very well those well-separated and defined clusters in the map. But t-SNE is a very dependent method of the type of individuals or objects that are sought to represent, wich also requires having a prior knowledge of the data, thus moving away from the unsupervised analysis.

Índice de Capítulos

Índice de Gráficos

Índice de Ilustraciones

1. Introducción

A lo largo de la historia de los seres humanos, los hitos de la civilización han estado marcados por avances en nuestra capacidad de observar y reunir información. El desarrollo de herramientas para medir la distancia, el peso o el tiempo; la utilización de datos sobre el firmamento para navegar por el mundo y abrir los profundos mares al comercio global; o el uso de datos para relacionar los brotes de cólera con el mal clima para salvar vidas serían algunos de los muchos ejemplos a destacar en este aspecto. En la mayoría de ocasiones los datos eran escasos, por lo que incluso cantidades limitadas de datos nos han brindado perspectivas clave para encontrar soluciones inesperadas a alguno de nuestros más exigentes desafíos, lo que pone de manifiesto la gran importancia que el uso de éstos ha tenido en el avance de la sociedad (1).

En el siglo XXI estamos experimentando un aceleramiento en este proceso. Los datos dejaron hace tiempo de ser un recurso escaso para convertirse en un recurso fundamental, renovable y cada vez más abundante. Por ejemplo, según datos de IBM, casi el 90% de la totalidad de datos existentes en el año 2016 habían sido creados en los últimos dos años anteriores; es decir, 2014 y 2015. Nuestra capacidad para conectar varios dispositivos y sensores a Internet hace que hoy día se generen cantidades masivas de datos a velocidades exponenciales. Se trata de la llamada era de los datos o *Big Data*.

El término *Big Data* hace referencia a la gestión y análisis de grandes volúmenes de datos que no pueden ser tratados de manera convencional (2). Su objetivo, al igual que los ya conocidos sistemas analíticos, es convertir el dato en información que facilite la toma de decisiones, y es en este apartado donde entra en juego el ingenio humano.

Los datos no tienen valor por sí solos. Éstos únicamente tienen valor cuando son comprensibles, por lo que el papel de los científicos o analistas de datos es indispensable para poder interpretarlos y obtener información útil de la gran cantidad existente en la actualidad. Las habilidades humanas son muy útiles a la hora de descubrir correlaciones desconocidas, patrones ocultos u otro tipo de información, aunque también deben ser ayudadas o acompañadas de las herramientas adecuadas.

En la era *Big Data,* una de las herramientas más utilizadas por los analistas es la **visualización de los datos**. Gracias a ella, los datos pueden ser presentados de forma más sencilla de entender, destacando las tendencias y los valores erróneos, o resaltando la información útil contenida en los mismos. Se trata de una herramienta muy importante en nuestros días, puesto que actúa independientemente del volumen de los datos, permitiéndonos hacer interpretaciones sobre grandes

cantidades de datos que de otra forma resultarían imposibles de procesar y muy costosas (en tiempo y recursos) de abordar para su análisis.

Otras de las herramientas o técnicas enfocadas también a trabajar con abundantes cantidades de datos y características son las conocidas como de **reducción de la dimensión**. Lo que hacen estas técnicas no es más que realizar una disminución del número de variables contenidas en un conjunto de datos, lo que facilita en gran medida el análisis de éstos. Además, y aunque no es el objetivo principal de estos métodos, permiten también realizar representaciones o visualizaciones de los datos, muy útiles a la hora de buscar estructuras en el análisis de cualquier conjunto.

Se trata de dos herramientas que a día de hoy no solo son muy utilizadas, sino que en muchos casos van de la mano. Tal es la conexión entre ambas que se han creado diversas técnicas que están enfocadas a la representación o visualización de datos, previa reducción de su dimensión. Este es el caso de **t-Distributed Stochastic Neighbor Embedding**, una técnica probabilística no lineal de reducción de la dimensión cuyo principal fin es el de proporcionar una visualización de los datos. Sin embargo, a pesar de ser un método reciente y bastante aclamado por sus usuarios, no existe mucha documentación acerca de él. Por tanto, el objetivo de este trabajo no es otro que el de analizar detalladamente dicha técnica, comprobando así si la fama de la que goza se encuentra en consonancia con los resultados que pueden obtenerse al aplicarla de diversos modos.

Así, el presente documento comenzará recapitulando y describiendo de manera breve algunas de las diversas técnicas de visualización y reducción de la dimensionalidad que pueden encontrarse en la literatura, lo que nos servirá para asentar los precedentes a la hora de hablar y explicar la técnica que aquí se analiza. Este análisis se llevará a cabo en los capítulos posteriores, en los que también se hablará de su antecesora, Stochastic Neighbor Embedding, algo que resulta indispensable para poder desarrollarla con minuciosidad. Por último, y una vez comprobadas las distintas aplicaciones de la técnica, se mostrarán los posibles inconvenientes que el usuario de la misma podría encontrarse.

2. Antecedentes de t-SNE

Una vez hablado sobre la importancia tanto de la visualización previa de grandes volúmenes de datos como de la reducción de la dimensión de los mismos, y antes de entrar de lleno en la materia principal que en este trabajo nos atañe, es preciso comenzar describiendo brevemente algunas de las más conocidas y populares técnicas pertenecientes a cada uno de esos campos, lo que nos permitirá

saber de dónde parte y en qué se diferencia la principal técnica presentada en este documento con todas aquellas que serán reseñadas en este apartado.

2.1. Técnicas de visualización

Como ya se comentó en la introducción, la visualización de los datos y su exploración visual juegan un papel importante en todos los procesos de análisis. Los analistas necesitan herramientas para crear hipótesis sobre conjuntos de datos complejos (es decir, aquellos que son muy grandes y/o contienen muchas dimensiones), y es aquí donde nacen las técnicas de visualización. Gran parte de estas técnicas están enfocadas al análisis exploratorio de los datos, aunque también existen otras muchas que permiten la visualización de modelos. Este apartado se centrará en las primeras mencionadas, puesto que son aquéllas que permiten identificar patrones interesantes y anteriormente desconocidos sin tener que acudir a la construcción de modelo previo alguno.

Las técnicas visuales utilizadas para la exploración de los datos se encuentran recogidas en diversos grupos o clases, según correspondan a un principio básico de visualización. Así, pueden encontrarse técnicas de todo tipo: **geométricas**, **iconográficas**, **basadas en píxeles** y **jerárquicas o apiladas** (3).

Las **técnicas geométricas** tienen como objetivo encontrar transformaciones “interesantes” de conjuntos de datos multidimensionales. En ellas, todas las dimensiones son tratadas igualmente, y son muy efectivas a la hora de detectar datos anómalos y correlaciones entre las diferentes dimensiones. Ejemplos pertenecientes a esta familia de técnicas podrían ser las conocidas *matrices de dispersión*, la *visualización radial de coordenadas* (más conocido como *Radviz*), o *Parallel Coordinates*.

Otra clase de técnicas usadas para la exploración visual son las **iconográficas**. Éstas suelen emplearse más en conjuntos con no una muy elevada dimensionalidad, ya que su interpretación no es tan directa. Busca asignar los valores de los atributos de un elemento multidimensional a las características de un icono, como estrellas o pequeñas caras, por ejemplo. *Color Icons*, *Stick Figure* o *Star Glyphs* formarían parte de este grupo de técnicas.

Las **técnicas basadas en píxeles** descansan en una idea similar a las iconográficas, pero en su lugar asignan el valor de la dimensión a un píxel de color y agrupa los píxeles pertenecientes a cada dimensión en áreas adyacentes. *Circle Segments* y *Recursive Pattern* son las técnicas más conocidas dentro de este grupo.

Por último, existe un grupo de técnicas que se adaptan para presentar los datos particionados de una manera **jerárquica**. Éstas incrustan dimensiones dentro de otras dimensiones en la representación, lo que las hace ajustarse mejor a conjuntos de datos de pequeña o mediana dimensión para no dificultar la interpretación al usuario. En este grupo destacan las técnicas *Dimensional Stacking*, *Cone Trees* y *Treemap*.

A continuación, se describirán y desarrollarán de manera breve las técnicas más conocidas de cada una de estas clases, para así poder también profundizar más en su funcionamiento:

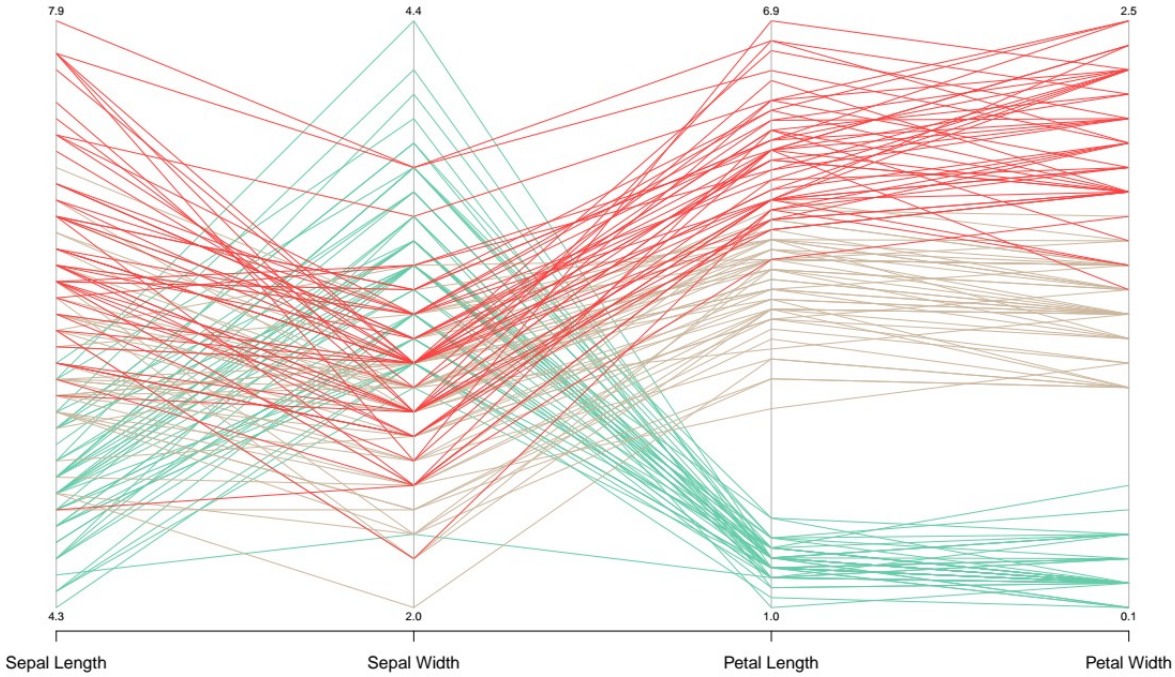
Parallel Coordinates

Parallel Coordinates es una forma común de visualizar geometría de alta dimensión y analizar datos multivariados. Esta visualización está estrechamente relacionada con la visualización de series temporales, salvo que se aplica a datos donde los ejes no corresponden a puntos en el tiempo, por lo que no tienen un orden natural.

En un gráfico de *Parallel Coordinates*, cada variable o dimensión tiene su propio eje y todos ellos se colocan de manera paralela entre sí. Cada uno de estos ejes puede tener diferente escala, ya que las características a las que representan pueden trabajar con distintas unidades de medida. Después, los valores de las diferentes variables que se correspondan a cada individuo del conjunto se trazan como una serie de líneas que se conectan entre todos estos ejes, lo que significa que cada línea es una colección de puntos colocados en cada eje.

Uno de los problemas de esta técnica es que la representación puede saturarse demasiado ante la existencia de muchos individuos. Sin embargo, suele ser muy útil para comparar muchas variables y buscar relaciones entre ellas.

Gráfico 1: Ejemplo de Parallel Coordinates.



El gráfico de arriba corresponde al conocido conjunto de datos Iris, el cual contiene información acerca de distintas medidas relacionadas con tres tipos de plantas iris: setosa, versicolor y virgínica. Gracias a este gráfico de *Parallel Coordinates*, vemos como al menos dos de los tres tipos pueden ser rápidamente diferenciados por sus medidas. Las plantas de tipo setosa se caracterizan por tener un pétalo corto y estrecho, y un sépalo también corto, aunque bastante ancho, mientras que las de tipo virgínica presentan unas características totalmente contrarias.

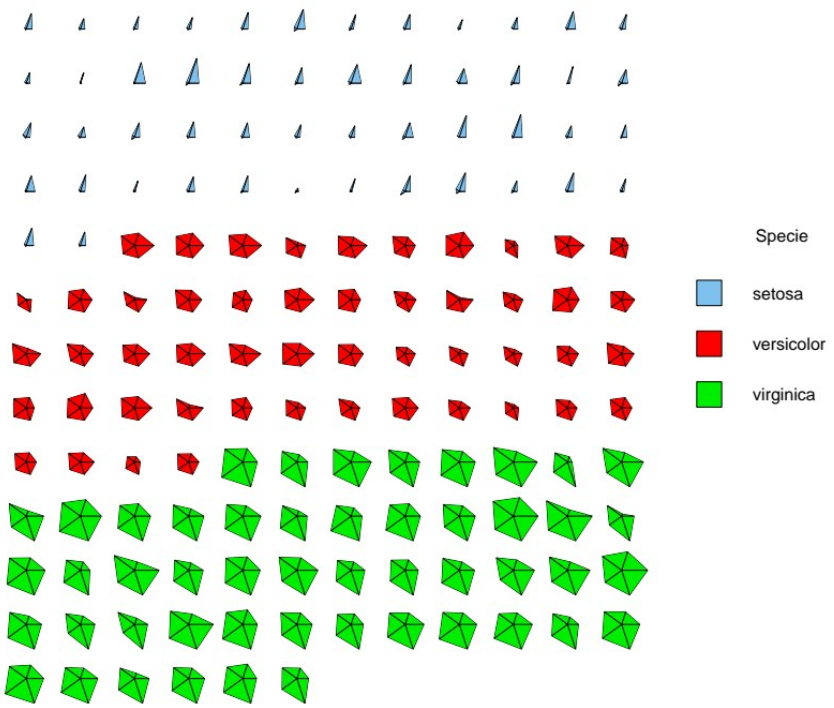
Star Glyph

Star Glyph es una de las más ampliamente usadas técnicas de visualización iconográficas o basadas en glifos aplicada a datos multivariantes. Al igual que otras técnicas pertenecientes a esta clase, proporciona una representación de los datos donde el conjunto es presentado como una colección de objetos visuales, en este caso, estrellas.

La visualización de un glifo de estrella representa un conjunto de datos en el que cada estrella corresponde a un registro de datos independiente (4). Cada variable o dimensión se representa por una rama o radio dentro de esa estrella que parte desde el origen, y cuya longitud se corresponde al valor de la variable en cuestión. Así, registros que compartan características similares presentarían

ramas del mismo tamaño, hasta el punto de que si son idénticos en todos los aspectos llegarían a tener la misma forma.

Gráfico 2: Ejemplo de Star Glyph.



En este ejemplo se representa de nuevo el conjunto de datos Iris antes mencionado. Al igual que la anterior técnica, esta basada en glifos nos permite ver la gran diferencia existente en las medidas entre las plantas de los dos tipos también antes mencionados, setosa y virgínica, si bien en este caso no sería posible conocer a simple vista en qué medidas existe esa discrepancia exactamente.

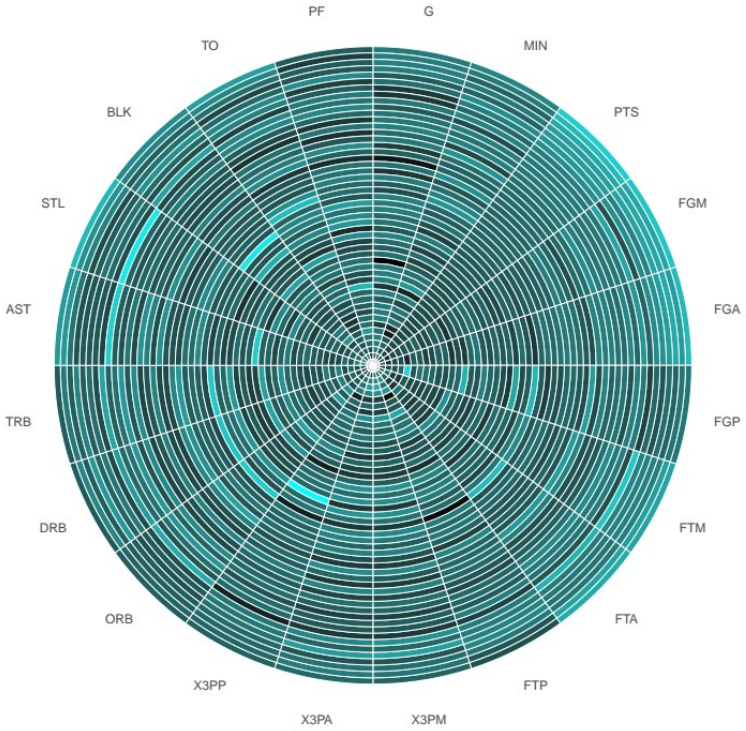
Circle Segments

Circle Segments es una técnica muy potente a la hora de visualizar grandes cantidades de datos, proporcionando unas visualizaciones más expresivas y vistosas que otras técnicas más conocidas, como, por ejemplo, la ya vista *Parallel Coordinates*.

Se trata de una técnica de visualización orientada a píxeles, cuya idea básica consiste en representar los datos en un círculo dividido en tantos segmentos como variables haya. Dentro de cada segmento se muestra el valor de la variable con un colorido píxel de acuerdo a su escala. Así, si el valor de los datos es elevado, el color del píxel será claro y brillante, mientras que, si el valor escalado es bajo, será oscuro. La disposición de los píxeles comienza en el centro del círculo y continúa hacia el

exterior colocando una línea ortogonal al segmento divisor. Así, los valores próximos al centro del círculo son próximos entre sí, potenciando la comparación visual de los valores (5).

Gráfico 3: Ejemplo de Circle Segments.



Este ejemplo de *Circle Segments* representa un conjunto de datos en el cual se pueden encontrar datos referentes a las actuaciones por partido de los 50 mayores anotadores de la National Basketball Association (NBA) en la temporada 2008 – 2009. Este tipo de gráficos también permiten ver rápidamente relaciones entre variables. Por ejemplo, vemos como aquellos jugadores que juegan más partidos, a su vez juegan más minutos y anotan más puntos, algo totalmente esperable.

Treemap

Treemap es una técnica de visualización que es capaz de representar y mostrar una gran cantidad de datos jerárquico. Este método suele ser utilizado para mostrar dos tipos de información al mismo tiempo: como está dividido el conjunto de datos y como está organizada la jerarquía. No obstante, puede también ser utilizada en aquellos casos en los que no existe jerarquía alguna en los datos (6).

Los datos son representados a través de varios rectángulos anidados que varían en tamaño y color según el valor de la variable o dimensión a la que correspondan. Así, a cada variable o categoría se le asigna un rectángulo que a su vez contiene dentro otros rectángulos pertenecientes a sus

subcategorías, de manera que el tamaño del área de la categoría o variable principal será el total de las subcategorías.

Aunque su principal desventaja es que no muestra los niveles jerárquicos de una manera muy clara, sigue siendo una opción muy eficiente, puesto que brinda una visión general muy rápida de la estructura del conjunto.

Gráfico 4: Ejemplo de Treemap.

El *Treemap* que puede verse justo antes representa la población de los diferentes continentes en los que podríamos distribuir la geografía del mundo. Como se comentaba en la descripción de la técnica, ésta es muy útil sobre todo a la hora de observar la estructura de un conjunto. En este caso, el gráfico nos permite ver como el continente asiático es el que cuenta con mayor población entre todos y, a su vez, como el sur y el este de ese continente comprenden las áreas más pobladas dentro del mismo.

2.2. Técnicas para la reducción de la dimensión

De igual modo que las técnicas de visualización, también existen técnicas de reducción de la dimensión de los datos que permiten representar y visualizar en dos o tres dimensiones aquellos conjuntos de datos que presentan muchas variables o atributos. Sin embargo, y a diferencia de las

técnicas recapituladas en el apartado anterior, éstas no solo son utilizadas para la representación, sino que también son útiles a la hora de aplicar algún modelo o algoritmo sobre los datos.

La mayoría de las técnicas de reducción de la dimensionalidad suelen crear nuevas variables como combinación de todas las existentes en el conjunto, con el objetivo de que estas nuevas mantengan la misma información. Usar estas nuevas variables para ajustar un modelo no solo hará disminuir el coste computacional y el tiempo de ejecución del algoritmo, también facilitará la interpretación al analista, al tener que lidiar con un número mucho menor de atributos.

Estas técnicas pueden dividirse en dos grandes grupos. Por un lado, se encuentran las tradicionales **técnicas lineales**, las cuales son muy efectivas cuando, al representar en un gráfico o mapa los distintos datos del conjunto a través de puntos, éstos forman una estructura lineal en el espacio de entrada. En caso de que esos puntos estén dispuestos a lo largo de superficies altamente plegadas o curvadas, debería utilizarse alguna técnica perteneciente al otro grupo: las **técnicas no lineales** (7).

Nuevamente, se presentarán, a modo de ejemplo, aquellas técnicas más populares y conocidas de cada uno de los grupos antes mencionados:

Técnicas lineales

Análisis de Componentes Principales (PCA)

El *Análisis de Componentes Principales* (en inglés, PCA) es una técnica lineal de análisis no supervisado que simplifica la complejidad de los datos de alta dimensión al tiempo que conserva su variabilidad. Permite mostrar gráficamente las posiciones relativas de los datos en menos dimensiones reteniendo la mayor cantidad de información posible, y explorar las distintas relaciones entre las diferentes variables (8).

PCA realiza una rotación de los datos, pasando a un conjunto de nuevas variables o dimensiones llamadas éstas *componentes principales* (PCs), las cuales no son más que combinaciones lineales de las originales (9) y retienen o capturan la mayor parte de la varianza de los datos.

Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) es un popular enfoque estadístico que permite encontrar y explorar dimensiones subyacentes con el fin de explicar la similitudes o diferencias entre los conjuntos de datos investigados. De forma similar al PCA, esta técnica crea un mapa en el cual se muestran las posiciones relativas de varios objetos, pero difiere de ésta puesto que solo es necesario conocer la distancia o alguna otra medida de similitud entre los objetos para su representación (10).

El objetivo que pretende el algoritmo de MDS es el de representar esa matriz de distancias o disimilitudes mediante un conjunto de variables llamadas *coordenadas principales*, de manera que las distancias entre las coordenadas de los elementos respecto a estas variables sean iguales o lo más próximas posibles a las de la matriz original (11).

Técnicas no lineales

Locally Linear Embedding (LLE)

Locally Linear Embedding (LLE) es un método no supervisado y no lineal para la reducción de la dimensión que puede descubrir estructuras no lineales en el conjunto de datos, y también preservar las distancias dentro de los vecindarios locales (12).

LLE obtiene una representación de los datos en la baja dimensión al suponer que incluso si los datos de alta dimensión forman un espacio no lineal, ésta aún puede seguir siendo considerada localmente lineal si cada punto y sus vecinos se encuentran en o cerca de un área localmente lineal de la variedad. Así, el algoritmo primero busca los vecinos más próximos al punto, gracias a los que calcula unos pesos o ponderaciones que permiten reconstruir nuevamente el punto y obtener unas coordenadas óptimas que serían representadas en menores dimensiones (13).

Maximum Variance Unfolding (MVU)

Maximum Variance Unfolding (MVU, o también conocido como *Semidefinite Embedding*, SDE) es una técnica no lineal de la reducción de la dimensión de los datos de alta dimensión que puede ser vista como una generalización no lineal del anteriormente descrito PCA. Este método se basa en una heurística simple: maximizar la varianza global y preservar las distancias locales entre las observaciones vecinas (14).

Al igual que otras técnicas anteriormente vistas (por ejemplo, MDS), esta técnica puede aplicarse en casos en los que solo se disponga de información sobre las similitudes locales. El algoritmo de MVU comienza buscando los vecinos más cercanos de cada punto en el espacio original. Para esos puntos vecinos, el algoritmo buscará otros nuevos para ser representados en menor dimensión, los cuales deben cumplir la restricción de preservar tanto las distancias como los ángulos originales. Además, los puntos obtenidos han de estar centrados en el origen. Por último, trata de “desplegar” los puntos o datos de la alta dimensión maximizando la varianza de su representación (15).

Curvilinear Component Analysis (CCA)

Curvilinear Component Analysis (CCA) es un método útil para la representación de estructuras de datos redundantes y no lineales. A diferencia de otras técnicas lineales o no lineales de reducción de la dimensión, ésta proporciona vistas curvilíneas incluso de estructuras fuertemente plegadas (16), puesto que se centra en preservar la topología local, definida aquí como la distancia entre pares.

En concreto, este método es una red neuronal auto-organizada cuyo principio es el de construir una relación entre los datos y la representación esperada a través de un conjunto de neuronas. Así, una vez tenidas en cuenta las distancias entre los diferentes puntos en la alta dimensión, la red buscaría mediante aprendizaje unos puntos que al ser representados en menos dimensiones preservasen esas distancias de la mejor manera posible.

3. t-Distributed Stochastic Neighbor Embedding (t-SNE)

Las diversas técnicas recopiladas y descritas en la sección anterior son solo algunos ejemplos de las muchas que pueden ser utilizadas a la hora de visualizar grandes conjuntos de datos o reducir la gran dimensión que presenten los mismos. No obstante, éstas no están exentas de problemas o inconvenientes.

En el caso de las técnicas de visualización, la mayoría solo proporcionan herramientas que muestran o representan más de dos o tres dimensiones, aunque su principal problema radica en el número de datos o individuos a ser representados, los cuales pueden dificultar mucho la interpretación del observador. Esto limita la aplicabilidad de las mismas, no pudiendo ser utilizadas sobre conjuntos de datos del mundo real, en los que existen miles de datos y elevadas dimensiones.

Los inconvenientes de las técnicas de reducción de la dimensión dependen del grupo al que éstas pertenezcan. Por ejemplo, las técnicas lineales más tradicionales se centran en representar alejados en la baja dimensión aquellos puntos que son más distintos, lo que las haría fallar en el caso de que los datos se encontrasen en una forma no lineal, donde es más importante mantener juntas las representaciones de los puntos más similares. Este problema sería solucionado mediante el uso de las técnicas no lineales, al tener como objetivo el preservar la estructura local de los datos. Pero es aquí donde nace la limitación de estas últimas, ya que hace que no sean capaces de captar la verdadera estructura global.

En esta sección se describe un método relativamente nuevo y popular utilizado para la visualización de conjuntos de datos de alta dimensionalidad conocido como **t-Distributed Stochastic Neighbor**

Embedding (t-SNE), capaz de capturar mucha de la estructura local de los datos, al tiempo que también revela su estructura global, tal como la presencia de grupos o clusters. Sin embargo, y antes de entrar en materia, es preciso comenzar hablando de la técnica de la que ésta precede, puesto que se trata de una mejora de la misma: **Stochastic Neighbor Embedding** (SNE).

3.1. Stochastic Neighbor Embedding

Como se comentaba anteriormente, la técnica presentada en esta sección es una mejora de **Stochastic Neighbor Embedding** (SNE de aquí en adelante) (17). SNE es un método probabilístico que trata de representar objetos de alta dimensionalidad en el plano, conservando la estructura de relación de la vecindad. La idea básica detrás de este algoritmo es la de minimizar una función de coste, la cual describe la divergencia entre la representación en la dimensión inferior y los datos reales. Esta función, por tanto, representa el error existente en la propia representación (18).

Esta técnica comienza convirtiendo la distancia Euclídea entre los puntos de la alta dimensión en probabilidades condicionadas que representan similitudes entre los diversos puntos.

Dado un conjunto de observaciones $\{\boldsymbol{x_1},\dots,\boldsymbol{x_n}\}$, donde $\boldsymbol{x_i}=(x_1,\dots,x_p)\,\forall\, i=1,\dots,n$, se define la probabilidad condicionada $p_{j|i}$. Esta es la probabilidad de que el punto $\boldsymbol{x_i}$ escoja al punto $\boldsymbol{x_j}$ como su vecino, en el caso de que los vecinos fuesen escogidos en proporción a su densidad de probabilidad bajo una distribución Gaussiana (o normal) centrada en el punto $\boldsymbol{x_i}$:

$$p_{j|i}=\frac{\exp\left(-\|\boldsymbol{x_i}-\boldsymbol{x_j}\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|\boldsymbol{x_i}-\boldsymbol{x_k}\|^2/2\sigma_i^2\right)}.$$

Así, cuanto más lejos esté un punto de $\boldsymbol{x_i}$, menor será la probabilidad de que lo escoja como su vecino y pertenezcan al mismo grupo.

El objetivo es representar cada punto i mediante una variable bidimensional $\boldsymbol{y_i}=(y_1,y_2)\,\forall\, i=1,\dots,n$, de forma que las posiciones entre los $\boldsymbol{x_i}$ se conserven en el plano. Por tanto, también es posible calcular una probabilidad condicionada para las contrapartes de los puntos $\boldsymbol{x_i}$ y $\boldsymbol{x_j}$ en el espacio de baja dimensión, denotada ésta como $q_{j|i}$:

$$q_{j|i}=\frac{\exp\left(-\|\boldsymbol{y_i}-\boldsymbol{y_j}\|^2\right)}{\sum_{k\neq i}\exp\left(-\|\boldsymbol{y_i}-\boldsymbol{y_k}\|^2\right)}.$$

Si los puntos representados en la dimensión inferior modelan correctamente la similitud entre los puntos del espacio de alta dimensión, estas probabilidades condicionadas serán iguales $(p_{j|i}=q_{j|i})$. Por tanto, tal y como se comentaba al principio de la sección, SNE tiene como objetivo encontrar una representación de los datos en la baja dimensión que minimice la disparidad entre estas probabilidades condicionadas.

Una medida de la fidelidad o exactitud con la que $q_{j|i}$ modela $p_{j|i}$ es la divergencia de Kullback-Leibler, la cual mide cómo de similar (o diferente) es una distribución de probabilidad respecto de otra (19). Esta divergencia tiene su origen en la teoría de la información, cuyo objetivo principal es el de cuantificar cuánta información hay en los datos. Por tanto, también se podría interpretar como una medida de la información perdida cuando se usa una distribución de probabilidad Q para aproximar otra P :

$$D_{KL}(P\|Q)=\sum_x P_x \log(\frac{P_x}{Q_x})$$

Esta medida de la similitud o diferencia siempre toma valores positivos, $D_{KL}(P\|Q)\geq 0$, obteniendo su valor mínimo si y solo si ambas distribuciones de probabilidad son iguales, lo que es fácilmente demostrable si comenzamos mostrando que $-D_{KL}(P\|Q)\leq 0$:

$$\begin{aligned}-D_{KL}(P\|Q)&=-\sum_x P_x \log(\frac{P_x}{Q_x})\\&=\sum_j P_x \log(\frac{Q_x}{P_x});\end{aligned}$$

sabiendo que $\log(a)\leq a-1$ para todo $a>0$:

$$\begin{aligned}-D_{KL}(P\|Q)&\leq \sum_x P_x(\frac{Q_x}{P_x}-1)\\&=\sum_x Q_x-\sum_x P_x\\&=1-1\\&=0.\end{aligned}$$

Resulta de interés comentar también el distinto tratamiento que la divergencia de Kullback-Leibler da a los dos casos más opuestos que nos podemos encontrar en cuanto a diferentes probabilidades.

En el caso en el que se aproxime una probabilidad P_x cercana al valor 0 mediante otra Q_x que tome un valor muy elevado, la divergencia tomará un valor de casi 0. Sin embargo, en el caso contrario (modelar una elevada probabilidad P_x mediante otra Q_x que tome un valor muy

cercano a 0), $D_{KL}(P_x||Q_x)$ será muy grande, ya que éstas serán muy diferentes. Esto pone de manifiesto la importancia que se le da a estos últimos casos, la cual estará también presente en SNE.

Entonces, esta divergencia correspondería con la función de coste a minimizar por el algoritmo de SNE:

$$C=\sum_i KL(P_i||Q_i)=\sum_i \sum_j p_{j|i}\log(\frac{p_{j|i}}{q_{j|i}}),$$

donde $P_i=(p_{j|i})_{j\neq i}$ representa la distribución de probabilidad condicionada sobre todos los otros puntos de datos dado el punto $\boldsymbol{x_i}$, , y $Q_i=(q_{j|i})_{j\neq i}$ representa la distribución de probabilidad condicionada sobre todos los otros puntos representados en el mapa dado el punto $\boldsymbol{y_i}$.

Observando la función anterior, puede verse como esta técnica trata de minimizar la suma de esta divergencia sobre todos los puntos de datos, puesto que las probabilidades condicionadas no son simétricas, es decir, $p_{j|i}\neq p_{i|j}$.

El parámetro restante a ser seleccionado es la varianza σ_i de la Gaussiana que está centrada sobre cada punto de alta dimensión $\boldsymbol{x_i}$. Lo más probable es que no exista un único valor óptimo de σ_i para todos los puntos del conjunto de datos, ya que la densidad de los datos probablemente varíe. En regiones densas, un valor más pequeño de la varianza es normalmente más apropiado que en regiones dispersas. Cualquier valor de σ_i induce a una distribución de probabilidad, P_i , sobre todos los otros puntos. Esta distribución tiene una entropía que se incrementa a medida que se incrementa su varianza.

Así, SNE lleva a cabo una búsqueda binaria para el valor de σ_i que produce una P_i con una *perplexity* fija que es especificada por el usuario. *Perplexity* (término en el que se profundizará más adelante) es definida como:

$$Perp(P_i)=2^{H(P_i)},$$

donde $H(P_i)$ es la entropía de Shannon de P_i medida en bits. La entropía de Shannon es una forma de medir la cantidad de información que contiene una variable aleatoria (20) y que, para el caso que nos atañe, tendría la siguiente forma:

$$H(P_i)=-\sum_j p_{j|i}\log_2(p_{j|i}).$$

La minimización de la función de coste se lleva a cabo mediante el **método del descenso del gradiente**. Se trata de un método muy utilizado en algoritmos de *Machine Learning*, el cual sencillamente mide cuánto cambia la salida de una función si cambia alguno de los parámetros del modelo, permitiendo así resolver problemas de minimización.

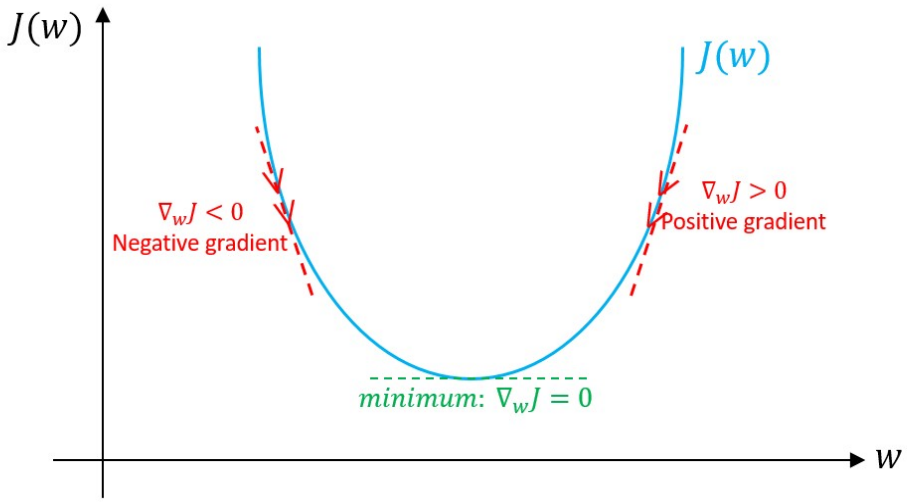
El **gradiente** es una generalización de la ya conocida derivada. Representa la pendiente en el punto en el que nos encontramos en la función de coste. Lo que hace este método en cada iteración es restar al valor del parámetro o parámetros de los que depende dicha función la derivada parcial respecto de los mismos, por lo que va actualizando tales parámetros, de manera que nos movamos por la función en dirección a la tangente en ese punto, pero siempre siguiendo una dirección que lleve hacia menores valores de la función (21).

Para entender mejor cómo funciona este método, podemos utilizar una función J dependiente de un solo parámetro w inicializado en un valor arbitrario. El descenso del gradiente actualizaría dicho parámetro según la siguiente regla:

$$w = w - \alpha \frac{\partial}{\partial w} J(w).$$

Por tanto, si la pendiente de la tangente es positiva, significaría que si aumentamos el valor del parámetro w también estaríamos incrementando J, por lo que deberíamos movernos en la dirección opuesta. Por el contrario, si la pendiente es negativa, al aumentar el parámetro disminuiríamos el valor de J, y esa es la dirección en la que queremos ir. Este funcionamiento se encuentra resumido de forma sencilla en la siguiente ilustración:

Ilustración 1: Funcionamiento del método del descenso el gradiente.



Fuente: https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3

Un parámetro muy importante en el funcionamiento del descenso del gradiente es α , el cual es conocido como *learning rate*. No obstante, debido también a la importancia de la que éste goza a la hora de representar visualizaciones, será un parámetro en el que se profundizará más adelante.

En este caso, nos interesa conocer cómo cambia la función de coste según cambien los puntos representados en el mapa o espacio de baja dimensión, por lo que el gradiente tendría la siguiente forma:

$$\frac{\partial C}{\partial y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j),$$

donde $\boldsymbol{y_i}$ e $\boldsymbol{y_j}$ se corresponderían con los puntos mapeados o representados en el espacio de más baja dimensión, siendo éstos las contrapartes de los puntos $\boldsymbol{x_i}$ y $\boldsymbol{x_j}$ de alta dimensión.

No obstante, aunque la técnica SNE construye buenas visualizaciones y, en algunas ocasiones, mejores incluso que algunas de las técnicas vistas en el apartado anterior, también consta de algunos problemas o desventajas.

El primero de ellos descansa en la difícil optimización de su gradiente. Éste involucra todas las probabilidades condicionadas para los puntos i y j . Si recordamos, para el cálculo de estas probabilidades se utiliza una distribución normal o Gaussiana, por lo que cada una de ellas está compuesta por exponenciales, lo que puede causar que el gradiente muestre comportamientos

inusuales de manera muy rápida y, por tanto, que el algoritmo tarde mucho tiempo en converger (22).

El segundo problema está relacionado con la distinta distribución de las distancias entre los puntos existente en los espacios de alta y baja dimensión. Este problema es conocido como "*crowding problem*", y hace referencia a que el área en el mapa o espacio de dimensión inferior disponible para situar los puntos medianamente distantes en la alta dimensión no será lo suficientemente grande, en comparación con el área disponible para acomodar los puntos más cercanos. Así, si se quisiera representar de manera precisa las pequeñas distancias en la baja dimensión, los puntos que se encuentran a una distancia moderada serían emplazados demasiado lejos en el gráfico y, por tanto, todos los puntos se "aplastarían" en la representación provocando una aglomeración.

Para solventar los inconvenientes que esta técnica presenta, surge *t-Distributed Stochastic Neighbor Embedding*, una mejora de la misma la cual únicamente realiza dos simples cambios en su algoritmo de funcionamiento que permiten obtener mejores representaciones en la baja dimensión.

3.2. t-Distributed Stochastic Neighbor Embedding

La técnica **t-Distributed Stochastic Neighbor Embedding** (t-SNE de aquí en adelante) (23), al igual que SNE, es una técnica probabilística no lineal de análisis no supervisado y reducción de la dimensión, utilizada principalmente para la exploración de datos y la visualización de conjuntos de datos de alta dimensionalidad. Al tratarse de una mera pero eficaz mejora, la idea básica detrás de este algoritmo sigue siendo la misma que la descrita cuando hablábamos sobre su predecesora SNE.

En primer lugar, esta técnica utiliza una función de coste alternativa a la presentada en la anterior subsección:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right).$$

En lugar de minimizar las divergencias entre todas las probabilidades condicionadas, se minimiza una única divergencia de Kullback-Leibler entre una distribución de probabilidad conjunta en el espacio de alta dimensión, P, y otra, Q, en el espacio de baja dimensión. Esto es debido a que utiliza una versión simétrica de SNE (24), ya que tiene la propiedad de que las probabilidades utilizadas cumplen la condición de ser simétricas ($p_{ij} = p_{ji}$ y $q_{ij} = q_{ji}$).

En este sentido, conviene también mostrar la definición de probabilidad conjunta que utilizaría esta versión de SNE, tanto en el caso del espacio de menor dimensión

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq l} \exp(-\|y_k - y_l\|^2)},$$

como para el caso del espacio de alta dimensión

$$p_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma^2)}.$$

Sin embargo, esta última definición causaría problemas cuando un punto de alta dimensión $\boldsymbol{x_i}$ fuese un outlier. En tal caso, los valores de p_{ij} serían extremadamente pequeños para todo j, por lo que la localización del punto representado en la baja dimensión $\boldsymbol{y_i}$ tendría un efecto muy pequeño en la función de coste. Como resultado, la posición del punto mapeado no estaría bien determinada por la posición del resto de puntos representados. Este problema es solventado definiendo las probabilidades conjuntas p_{ij} como las probabilidades condicionadas "simetrizadas", esto es,

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}.$$

Esta definición asegura que cada uno de los puntos $\boldsymbol{x_i}$ tengan o hagan una contribución significativa a la función de coste.

Unido a esto, t-SNE también modifica la definición o función de probabilidad condicionada en el espacio de baja dimensión. El principal cambio que realiza sobre la definición de esta probabilidad condicionada es el de emplear una distribución t-Student con un grado de libertad (también conocida como distribución de Cauchy) para realizar el cálculo de las mismas. Usando esta distribución, las probabilidades conjuntas q_{ij} son definidas como

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

El uso de esta distribución soluciona los dos grandes problemas presentados por SNE. La distribución t-Student presenta una cola más pesada que la distribución Gaussiana, lo que permite que esos puntos aplastados referentes al "*crowding problem*" estén más dispersos, ya que una distancia moderada será representada fielmente por una distancia mucho mayor en el mapa.

Asimismo, el cambio de distribución (junto con el uso de la versión simétrica de SNE) también modifica el gradiente de la función de coste. Para poder obtener dicho gradiente a partir de la

divergencia de Kullback-Leibler antes mostrada, es útil definir dos variables auxiliares d_{ij} y Z previamente:

$$d_{ij} = \|y_i - y_j\|,$$

$$Z = \sum_{k \neq l} (1 + d_{kl}^2)^{-1}.$$

Así, si cambia $\boldsymbol{y_i}$, únicamente cambiarán las distancias d_{ij} y d_{ji} para todo j . Entonces, el gradiente de la función de coste con respecto a $\boldsymbol{y_i}$ viene dado por

$$\begin{aligned} \frac{\partial C}{\partial y_i} &= \sum_j \left(\frac{\partial C}{\partial d_{ij}} + \frac{\partial C}{\partial d_{ji}}\right)(y_i - y_j) \\ &= 2 \sum_j \frac{\partial C}{\partial d_{ij}} (y_i - y_j). \end{aligned}$$

La derivada $\frac{\partial C}{\partial d_{ij}}$ es calculada a través de la definición de la divergencia de Kullback-Leibler:

$$\begin{aligned} \frac{\partial C}{\partial d_{ij}} &= -\sum_{k \neq l} p_{kl} \frac{\partial (\log q_{kl})}{\partial d_{ij}} \\ &= -\sum_{k \neq l} p_{kl} \frac{\partial (\log q_{kl} Z - \log Z)}{\partial d_{ij}} \\ &= -\sum_{k \neq l} p_{kl} \left(\frac{1}{q_{kl} Z} \frac{\partial \left((1 + d_{ij}^2)^{-1}\right)}{\partial d_{ij}} - \frac{1}{Z} \frac{\partial Z}{\partial d_{ij}}\right). \end{aligned}$$

El gradiente $\frac{\partial \left((1 + d_{ij}^2)^{-1}\right)}{\partial d_{ij}}$ no es cero cuando $k = l$ y $l = j$. Por tanto, la derivada $\frac{\partial C}{\partial d_{ij}}$ sería

$$\frac{\partial C}{\partial d_{ij}} = 2 \frac{p_{ij}}{q_{ij} Z} (1 + d_{ij}^2)^{-2} - 2 \sum_{k \neq l} p_{kl} \frac{(1 + d_{ij}^2)^{-2}}{Z}.$$

Dándose cuenta de que $\sum_{k \neq l} p_{kl} = 1$, este gradiente puede simplificarse a

$$\begin{aligned} \frac{\partial C}{\partial d_{ij}} &= 2 p_{ij} (1 + d_{ij}^2)^{-1} - 2 q_{ij} (1 + d_{ij}^2)^{-1} \\ &= 2 (p_{ij} - q_{ij}) (1 + d_{ij}^2)^{-1}. \end{aligned}$$

Sustituyendo este término en la ecuación de $\frac{\partial C}{\partial y_i}$, se obtiene el gradiente final:

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

En este nuevo gradiente solo hay un parámetro que involucre exponenciales, que es la probabilidad conjunta en el espacio de alta dimensión (p_{ij}). Gracias a esto, la optimización es mucho más sencilla que en el caso del gradiente de SNE, puesto que el algoritmo converge de manera mucho más rápida al mínimo.

Aunque t-SNE presenta varias ventajas sobre otras técnicas de reducción de la dimensión y visualización de datos, este algoritmo también presenta algunas limitaciones:

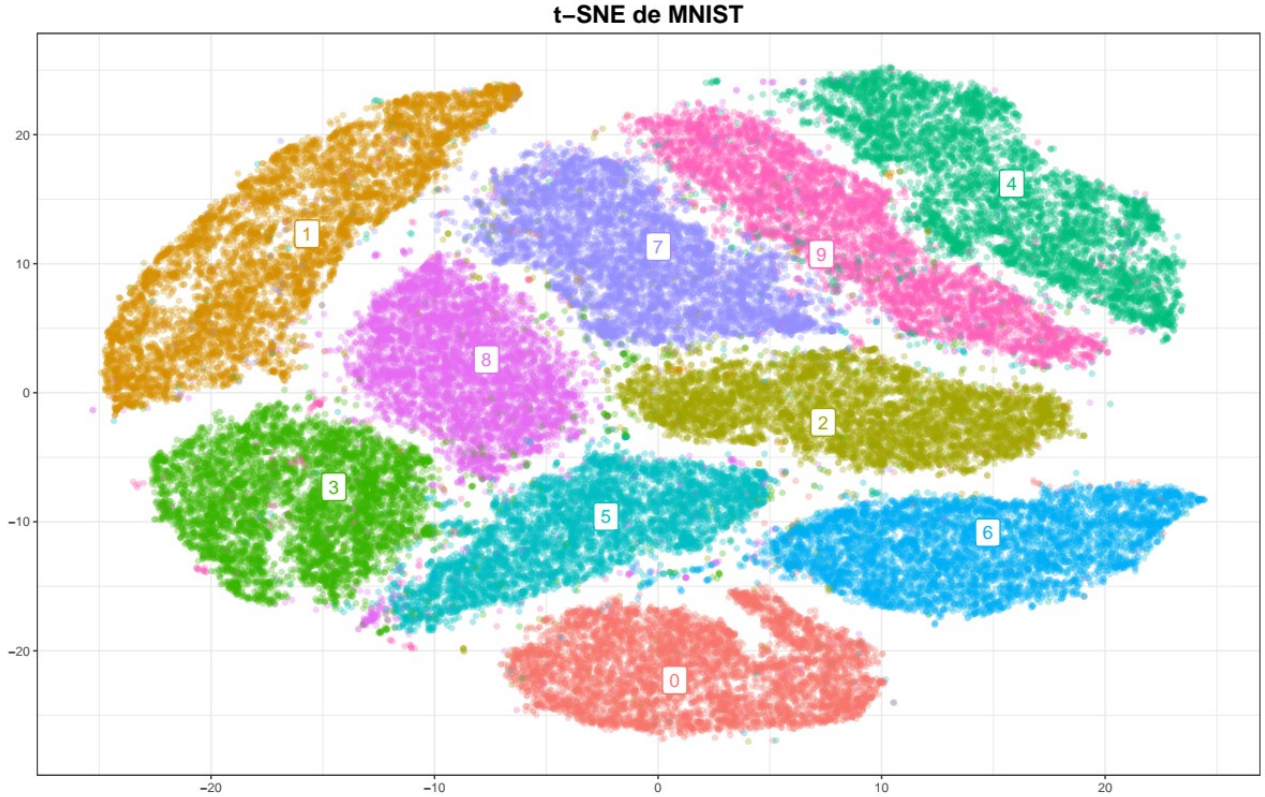
1. No está claro si t-SNE funcionaría en la reducción de la dimensión a más de 3 dimensiones. Esto se debe a que, en los espacios de alta dimensión, las colas pesadas comprenden una porción relativamente grande de la masa de probabilidad bajo la distribución t-Student, lo que puede conducir a representaciones de datos que no preserven su estructura local. Por eso, cuando se busque reducir a dimensiones más altas, sería más apropiado utilizar una distribución t-Student con más grados de libertad que uno.
2. t-SNE reduce la dimensión de los datos basándose principalmente en las propiedades locales de los mismos. En conjuntos de datos con alta dimensionalidad intrínseca, la suposición de linealidad local que realiza el algoritmo (al utilizar la distancia Euclídea) puede ser violada, por lo que no daría buenos resultados.
3. La no convexidad de la función de coste hace que existan varios mínimos locales, lo que requiere que se elijan varios parámetros de optimización que afectarían a la solución construida. Además, se trata de una técnica no determinista, por lo que podría ejecutarse varias veces con los mismos parámetros y obtener diferentes resultados.

A modo de ejemplo de las visualizaciones que proporciona esta técnica, se realizará una representación en dos dimensiones del conocido conjunto de datos MNIST, el cual contiene 60.000 imágenes en escala de grises de números escritos a mano, los cuales van de 0 a 9. Estas imágenes tienen 28 x 28 = 784 píxeles o, lo que para nosotros resulta más importante, dimensiones. No obstante, también resulta conveniente comparar este método con otros más conocidos y tradicionales, como es el caso del análisis de componentes principales (PCA), del cual ya se ha comentado su funcionamiento en la sección anterior.

Asimismo, debe reseñarse que tanto en esta como en las siguientes representaciones realizadas mediante la técnica t-SNE que este trabajo contenga, se comenzará reduciendo la dimensionalidad de los conjuntos a 50 dimensiones mediante PCA, como bien recomiendan los autores en su trabajo, en aras de acelerar el algoritmo y de que el coste computacional no sea excesivamente elevado.

Gráfico 5: Análisis de componentes principales sobre MNIST.

Gráfico 6: t-Distributed Stochastic Neighbor Embedding sobre MNIST.



Gracias a estos mapeados, puede verse rápidamente la diferencia entre ambos métodos. Es cierto que las dos primeras componentes principales del PCA logran mantener mucha de la información contenida en el conjunto de datos sobre las imágenes (especialmente para algunos de los dígitos), pero claramente no la suficiente para diferenciarlos todos, por lo que los concentra en un mismo grupo en el que es difícil distinguir los unos de los otros. Por su parte, el algoritmo de t-SNE nos facilita una visión mucho más nítida, en la que pueden distinguirse a simple vista los diferentes grupos que forman cada uno de los dígitos, encontrando además una clara separación entre los mismos.

Para finalizar con la explicación y comprensión de la técnica t-SNE, es preciso hablar de los dos parámetros fundamentales del algoritmo, los cuales influyen en gran medida en la construcción de las visualizaciones y deben ser seleccionados por parte del propio usuario: *perplexity* y *learning rate* (o tasa de aprendizaje).

El parámetro *perplexity*, como se ha visto anteriormente, está muy relacionado con la probabilidad condicionada en el espacio de alta dimensión. Éste puede ser interpretado como una suave medida del número efectivo de vecinos del punto en cuestión, el cuál se recomienda que tenga un valor comprendido entre 5 y 50. Fuera de ese rango, las representaciones podrían llegar a volverse un tanto extrañas.

Una buena manera de observar la influencia que ejerce este parámetro en las representaciones es realizar una comparación visual en la que se utilicen distintos valores del mismo. Para este ejemplo, se utilizará otro conjunto de datos llamado DIGITS, el cual, al igual que el anterior utilizado MNIST, contiene imágenes en escala de grises de números escritos a mano, pero que en este caso son de 8 x 8 = 64 píxeles (o dimensiones). Además, se trata de un conjunto de menor tamaño, conteniendo únicamente 1.797 imágenes:

Gráfico 7: t-SNE sobre DIGITS. Perplexity = 2.

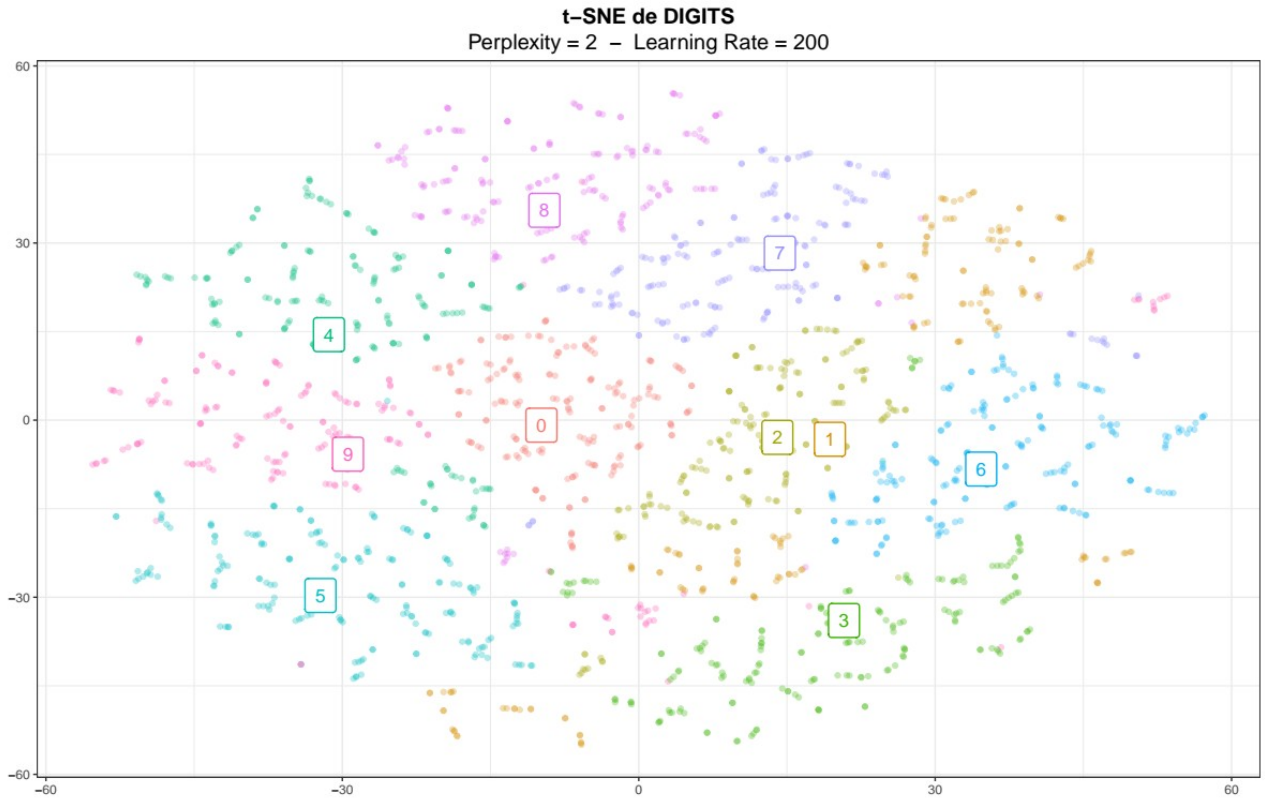


Gráfico 8: t-SNE sobre DIGITS. Perplexity = 30.

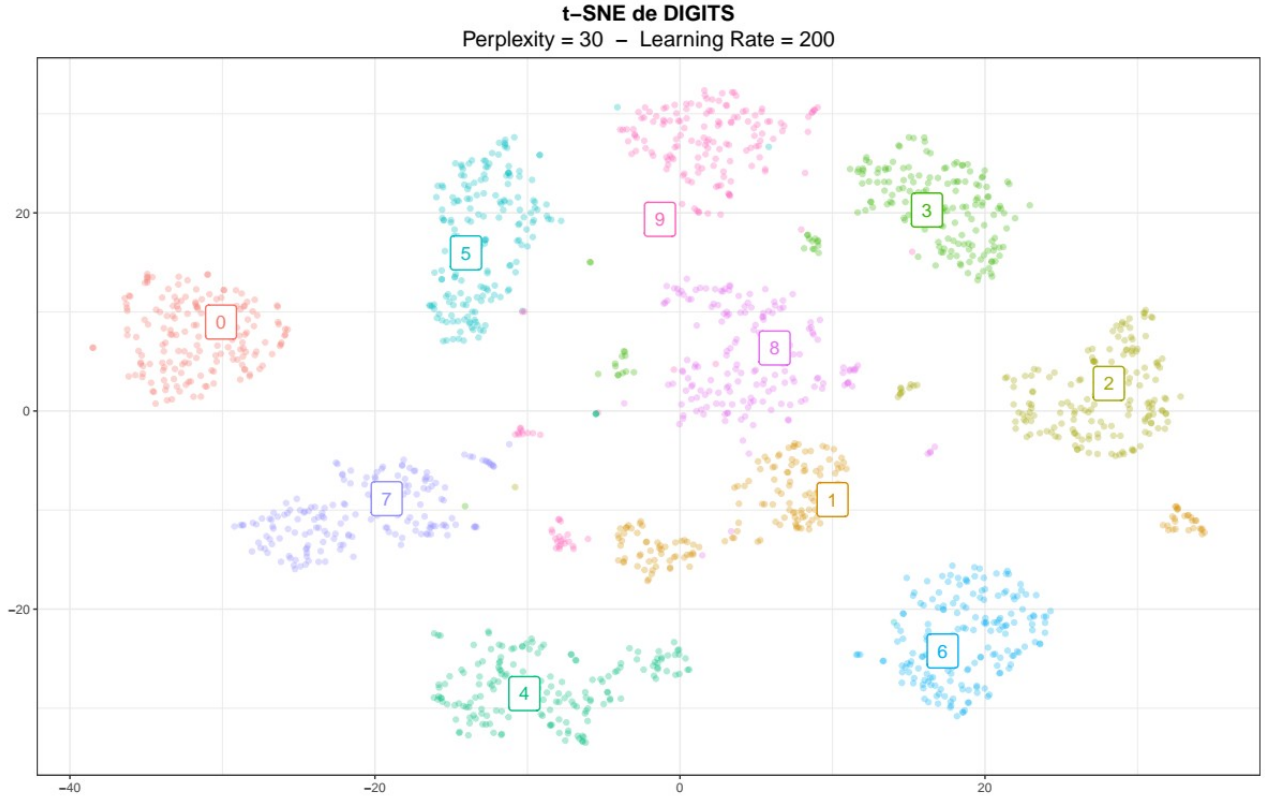


Gráfico 9: t-SNE sobre DIGITS. Perplexity = 300.

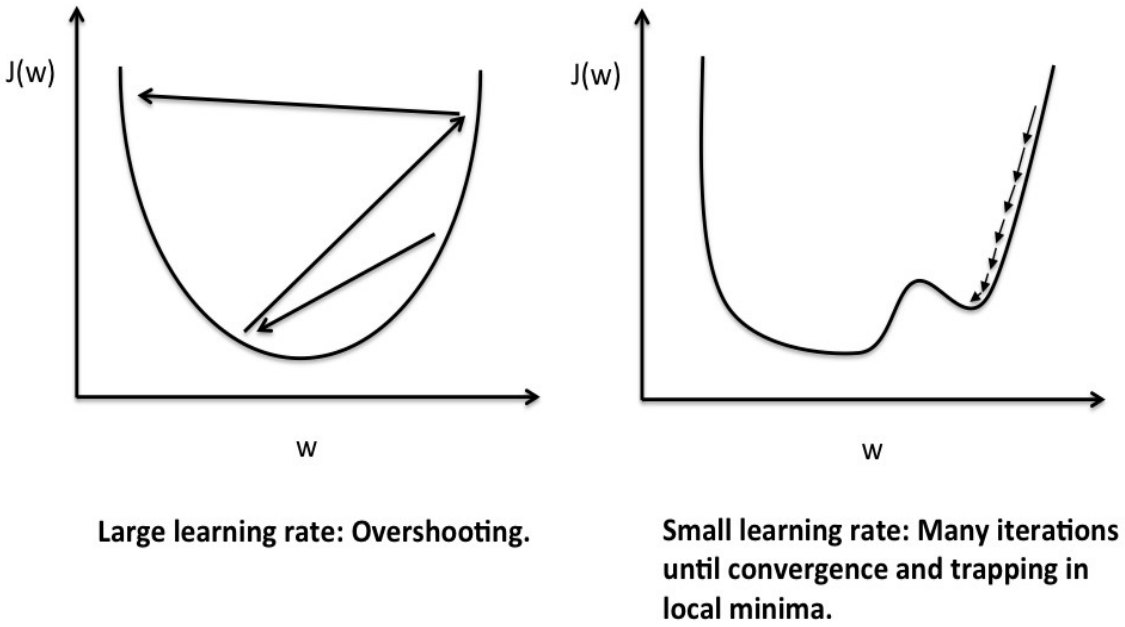
Vemos cómo el cambio en el valor del parámetro ha ido afectado mucho a la representación del conjunto de datos DIGITS en dos dimensiones. En el primer caso, en el cual se ha comenzado estableciendo un valor de *perplexity* muy bajo (Gráfico 7), los distintos grupos o clústers que forman los dígitos se encuentran bien definidos, aunque también es cierto que los puntos se encuentran muy dispersos entre sí. Asimismo, puede verse como existe algún que otro solapamiento y apenas espacio entre ellos. Aumentar el valor de *perplexity* a 30 (Gráfico 8) hace que aumente y mejore ese espacio entre los diferentes grupos, permitiendo ahora una distinción mucho más clara de los mismos. Por último, incrementar el valor del parámetro a uno tan exagerado como 300 (Gráfico 9) hace que muchos de los grupos comiencen a converger y a no distinguirse, como es el caso de los grupos que forman los dígitos 3, 5, 8 y 9.

Estas distintas visualizaciones ayudan a comprender el efecto de *perplexity* y la sensibilidad de este valor al resultado general. Así, un pequeño valor de *perplexity* puede llevar a mezclas más homogéneas de ubicaciones con muy poco espacio entre ellas, ya que son las variaciones locales las que dominan (25). Incrementar su valor separa los clústers de manera más efectiva, pero un valor excesivo hace que estos grupos se solapen o superpongan (26). Podría ser visto, por tanto, como un parámetro encargado de equilibrar los aspectos locales y globales del conjunto de datos.

La tasa de aprendizaje o *learning rate* es, por su parte, un parámetro crucial a la hora de minimizar la tan importante función de coste del algoritmo de t-SNE. Este parámetro controla la velocidad a la que el proceso es ejecutado, puesto que especifica el tamaño del paso del descenso del gradiente durante la optimización.

El valor de *learning rate* debe ser escogido cuidadosamente, ya que si este es muy pequeño, el modelo podría tardar mucho en converger a la solución óptima (si es que la encuentra) o incluso mantener la optimización en un mínimo local. Por otro lado, si el valor es demasiado elevado, el modelo podría no converger en una solución, pues éste saltaría por encima del punto más óptimo (Ilustración 2). Esto lo convierte en un parámetro muy influyente en la representación.

Ilustración 2: Funcionamiento de diferentes valores de learning rate en el descenso del gradiente.



Fuente: https://www.hackerearth.com/blog/developers/3-types-gradient-descent-algorithms-small-large-data-sets/

Para ver como afectan distintos valores de este parámetro en la visualización, realizaremos nuevamente una comparación visual, aunque esta vez utilizando los dos conjuntos de datos ya usados anteriormente. Comenzando con el conjunto DIGITS:

Gráfico 10: t-SNE sobre DIGITS. Learning rate = 5.

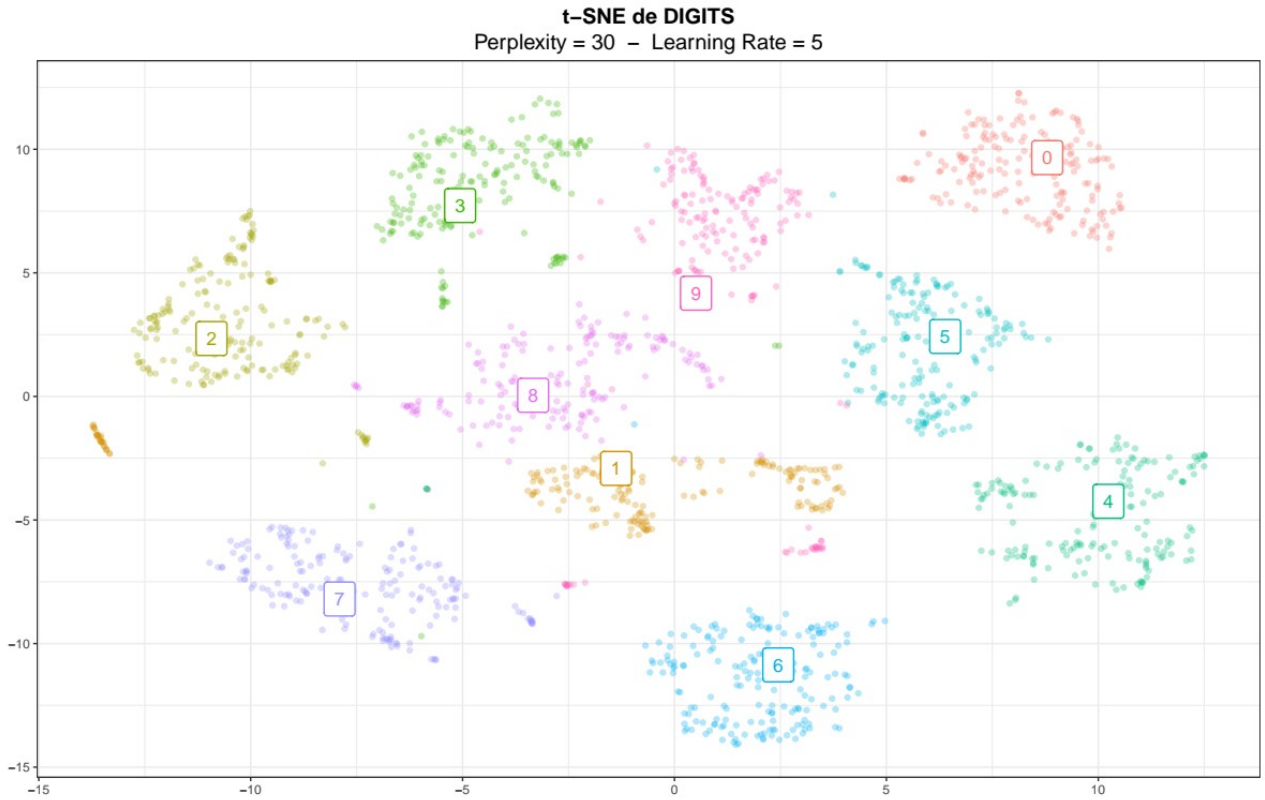


Gráfico 11: t-SNE sobre DIGITS. Learning rate = 50.

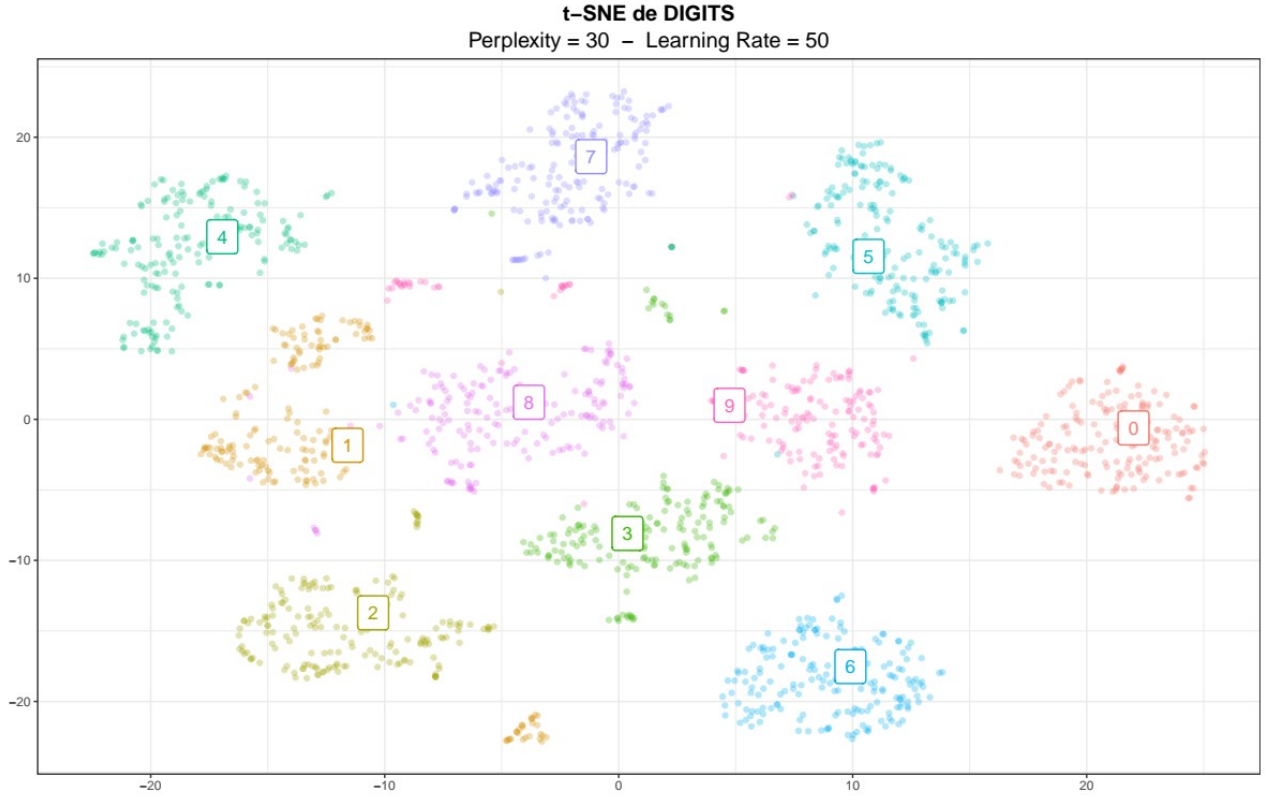


Gráfico 12: t-SNE sobre DIGITS. Learning rate = 800.

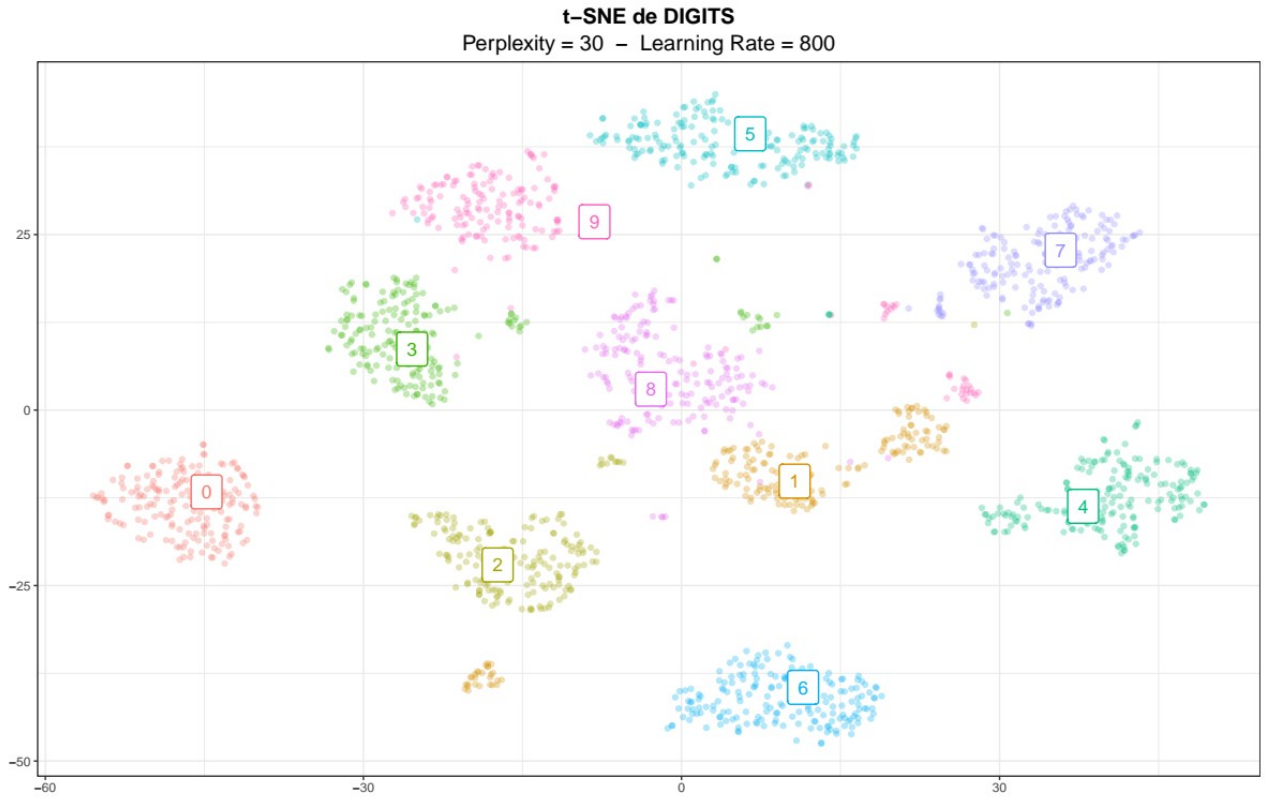
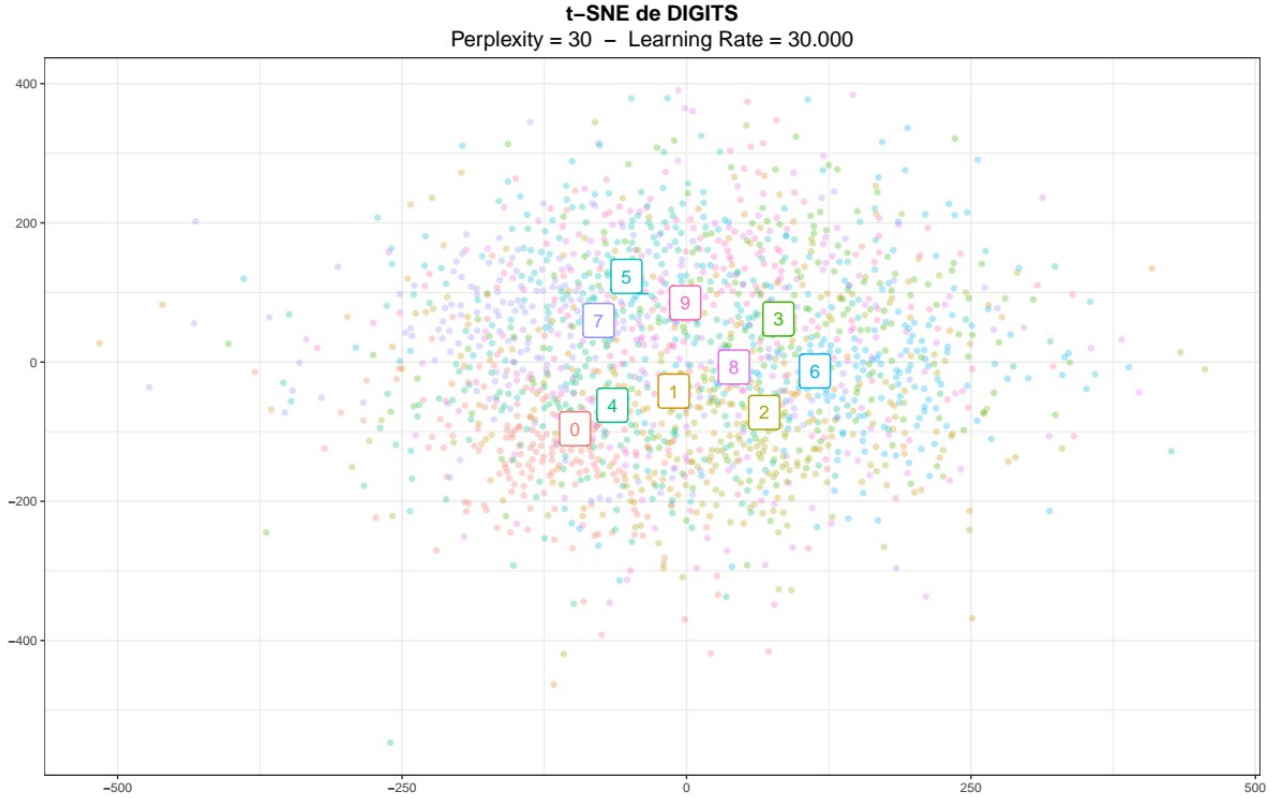


Gráfico 13: t-SNE sobre DIGITS. Learning rate = 30.000.



Como puede apreciarse, la visualización de los datos en el espacio de baja dimensión cambia mucho de un *learning rate* bajo a uno elevado. Al establecer un valor alto, los puntos se dispersan en el mapa sin patrón alguno, formando una especie de 'bola' en la que cualquier punto se encuentra aproximadamente equidistante de sus vecinos más cercanos y en la que es imposible distinguir los grupos que formarían las distintas imágenes de los dígitos. No obstante, en este caso, la decisión de escoger un valor tan bajo como 5 no ha afectado en gran medida a la visualización. Los grupos se encuentran muy bien separados y definidos, si acaso con una mayor dispersión de los puntos que forman cada uno de ellos.

Este resultado parece indicar que una baja tasa de aprendizaje no influye tanto como cabría esperar en aquellos conjuntos que podríamos denominar como no tan grandes; es decir, aquellos que no presentan un número demasiado elevado de dimensiones o que no están formados por tantos datos o individuos.

Si realizamos ahora la comparación sobre el conjunto MNIST podremos obtener más conclusiones:

Gráfico 14: t-SNE sobre MNIST. Learning rate = 5.

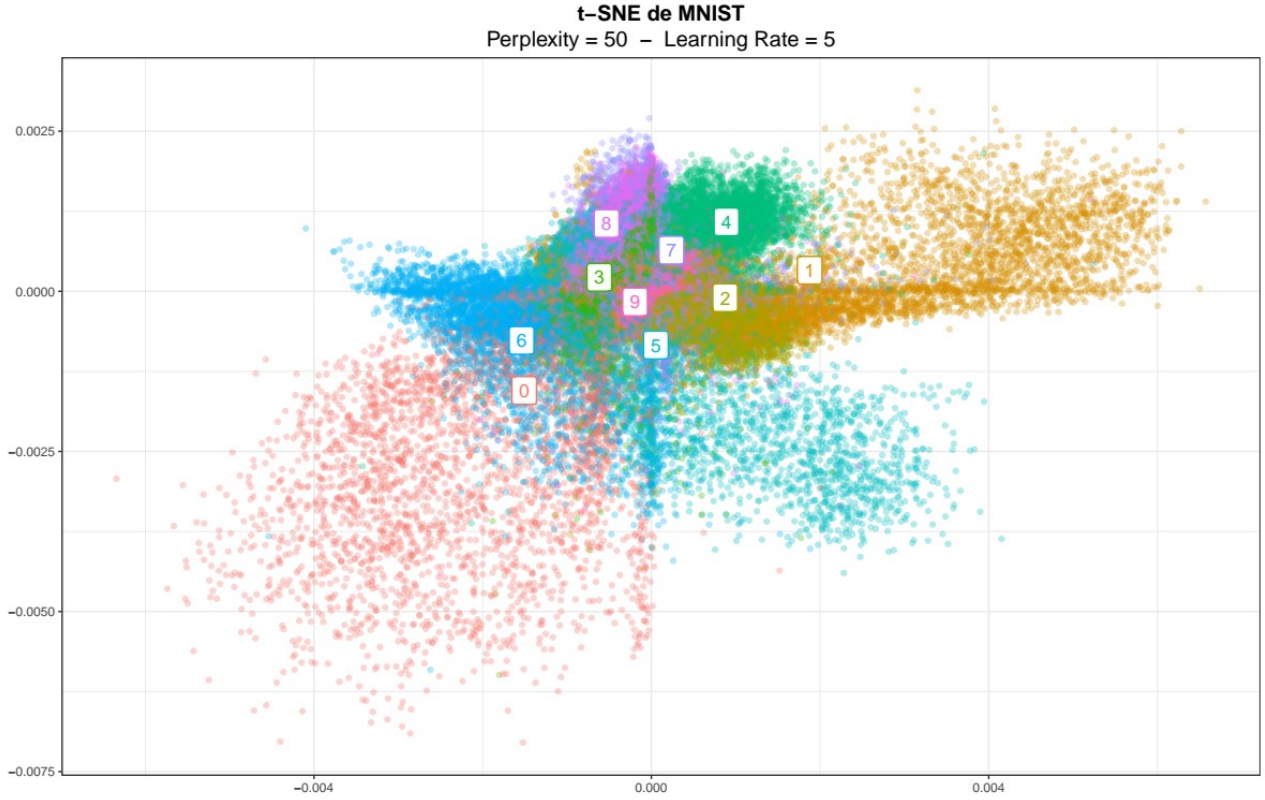


Gráfico 15: t-SNE sobre MNIST. Learning rate = 50.

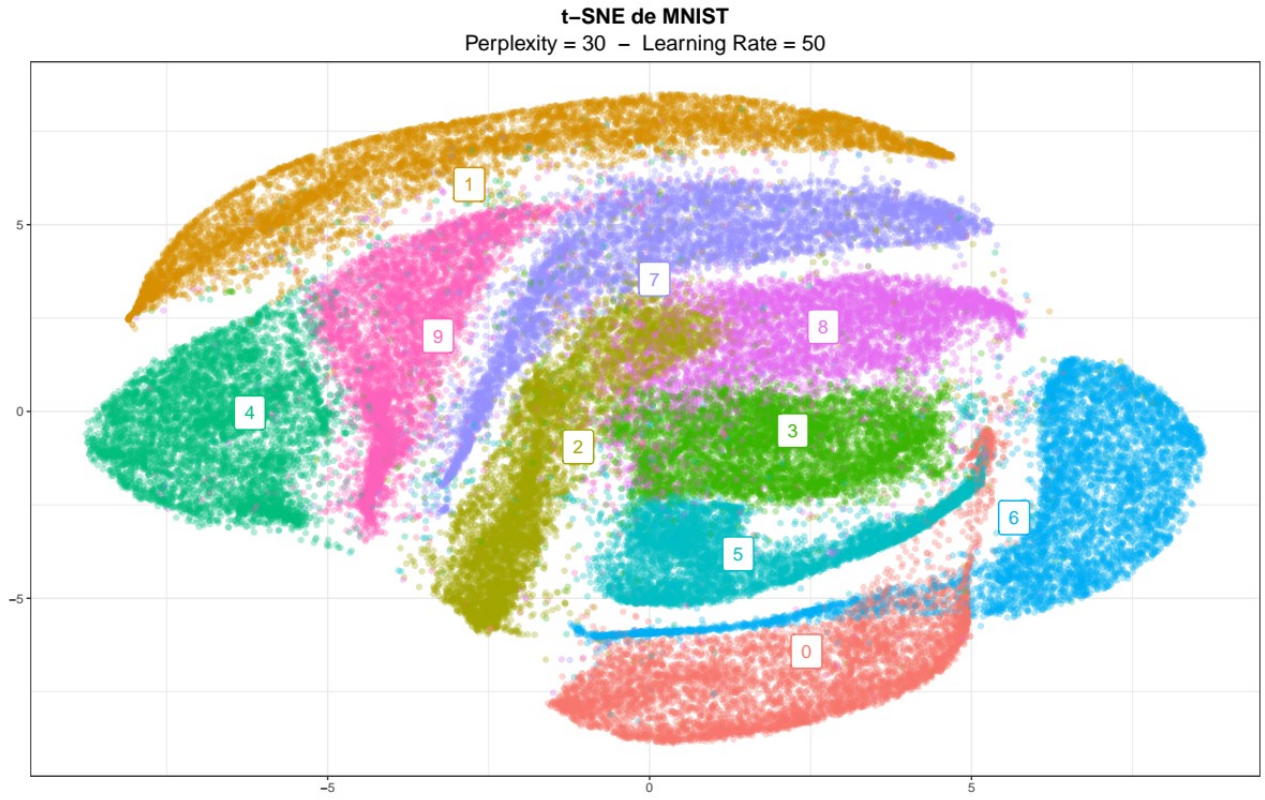


Gráfico 16: t-SNE sobre MNIST. Learing rate = 800.

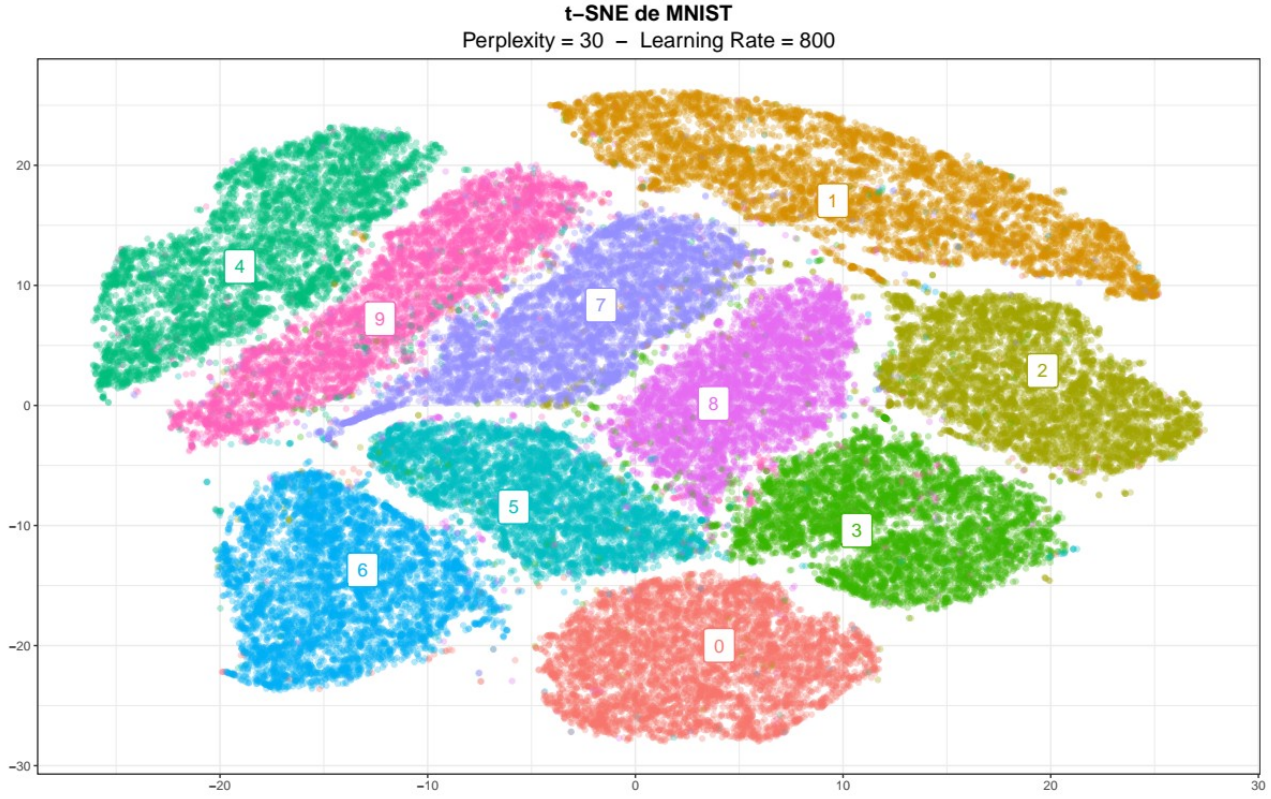
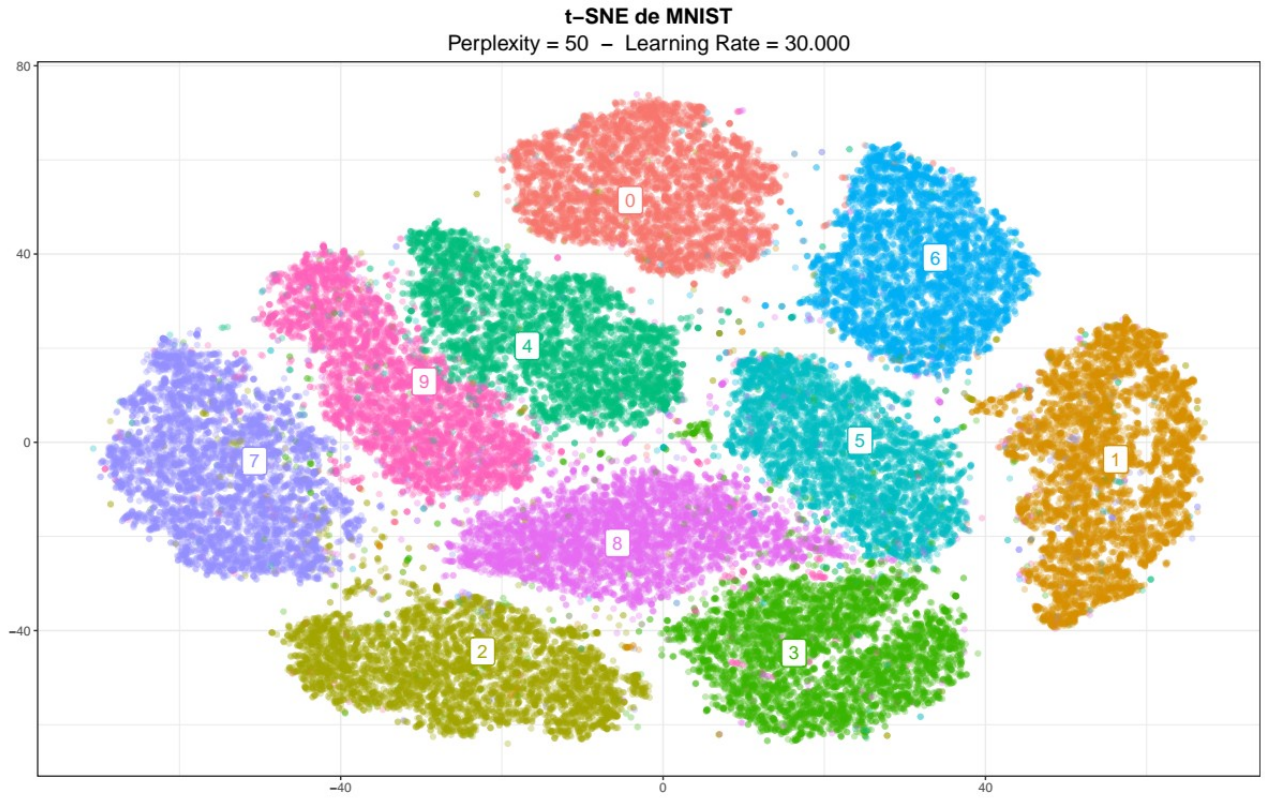


Gráfico 17: t-SNE sobre MNIST. Learning rate = 30.000.

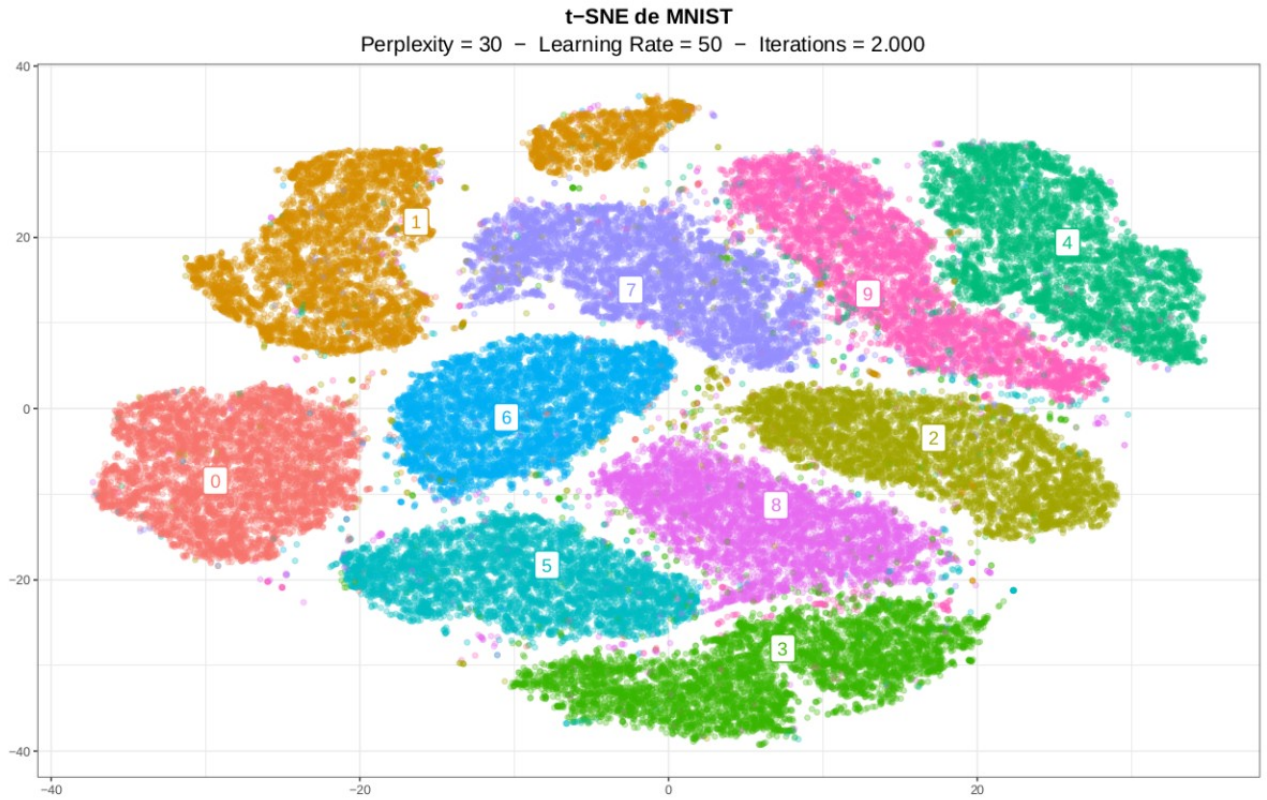


Utilizando ahora este conjunto de datos de mucho mayor tamaño que el anterior DIGITS, puede comprobarse cómo el *learning rate* influye sobre la representación de este tipo de conjuntos justo de la manera contraria: establecer un valor alto seguirá proporcionando una buena visualización de los datos, mientras que un valor pequeño haría que todos los puntos se comprimieran de una manera densa en el centro del mapa, al tiempo que algunos de ellos comienzan a separarse de esa especie de nube en la que resulta muy difícil hacer distinciones entre grupos. Además, se ve claramente como a medida que se va aumentando ese *learning rate* los grupos comienzan a ser mejor diferenciados.

No obstante, éste no parece ser un resultado razonable, en el sentido en el que una menor tasa de aprendizaje significa realizar una búsqueda más exhaustiva del mínimo de la función de coste. El problema está, entonces, en el número de iteraciones que debe realizar el algoritmo a la hora de buscar la solución óptima.

En ambos ejemplos (DIGITS y MNIST) se forzó al algoritmo a realizar unas 500 iteraciones. Si fuéramos más flexibles y permitiéramos a la técnica realizar las iteraciones necesarias, obtendríamos también una buena representación de aquellos conjuntos muy grandes de datos, incluso optando por escoger un valor pequeño de *learning rate*. Por ejemplo, si aumentásemos a 2.000 el número de iteraciones en el caso en el que se establece una tasa de aprendizaje de 50 para el conjunto MNIST, obtendríamos la siguiente representación:

Gráfico 18: t-SNE sobre MNIST. Learning rate = 50 - Iterations = 2.000.



Si bien esta nueva representación divide alguno de los grupos (los referentes a los dígitos 1 y 3), vemos cómo también interpreta fielmente la separación entre los diferentes clústers que forman las imágenes, al igual que en los casos en los que se utilizó un elevado *learning rate* y un menor número de iteraciones. Sin embargo, cabe destacar que existe un gran incremento en el coste computacional del algoritmo que quizás no se vea compensado por una mejora en la representación.

Así, una vez vista la diferencia en la influencia de la tasa de aprendizaje en la representación en la baja dimensión según el tamaño del conjunto de datos, podemos establecer que aquellos valores más elevados del parámetro afectan en mayor medida a los conjuntos más pequeños, en los que los que es más probable que grandes saltos del gradiente eviten encontrar el punto más óptimo. Por contra, los grandes conjuntos se verían más afectados por pequeños valores de *learning rate*, puesto que el algoritmo no gozaría del tiempo necesario para encontrar una solución óptima para la representación. Sin embargo, en caso de establecer un valor no muy elevado para este tipo de conjuntos, una solución podría pasar también por aumentar el número de iteraciones que debe realizar el algoritmo de t-SNE, para que así pueda disfrutar de más tiempo para encontrar el óptimo de la función de coste. Con todo, las recomendaciones suelen ser situar el valor de este parámetro dentro del rango 10 – 1.000, aunque como bien se ha mostrado, éste debe ir acorde al número de iteraciones.

Antes de finalizar con esta sección, cabe destacar que no existe ningún valor óptimo concreto para los parámetros *perplexity* y *learning rate* que ofrezcan mejores visualizaciones que otros, ya que, como hemos visto, éstos pueden variar en función de los datos, por lo que deben amoldarse a ellos. Lo más normal es encontrar dichos valores mediante un proceso de prueba y error.

3.3. Implementación en R

No obstante, si bien estos dos parámetros son los más importantes en cuanto a la construcción de visualizaciones, no son los únicos existentes y manejables dentro de la técnica t-SNE. Es por eso que, para al menos presentarlos y ver la influencia que estos ejercen en el algoritmo, aprovecharemos esta sección del documento para explicar la implementación de t-SNE en el software estadístico R, lo que también servirá de ayuda y de guía al propio lector en caso de querer probar por sí mismo la técnica en cuestión.

Cuando se habla de implementación, no se habla de una programación total iniciada desde cero, sino que hace referencia al uso de la función *Rtsne* disponible en la librería que lleva su mismo nombre (27). Esta función consta de muchos parámetros que pueden ser ajustados, aunque en este caso solo se aludirá a los más importantes dentro de ella:

```
Rtsne(X, dims = 2, perplexity = 30, theta = 0.5, max_iter = 1000, momentum =
      0.5, final_momentum = 0.8, eta = 200, exaggeration_factor = 12, …)
```

Los valores aquí mostrados de cada parámetro son los valores asignados por la propia función por defecto. *Perplexity* y *eta* (correspondiente al *learning rate*) son parámetros en los que ya profundizamos anteriormente y que no comentaremos nuevamente. Centrándonos ahora en el resto de ellos:

- *dims* alude al número de dimensiones a las que desea reducirse la dimensionalidad del conjunto en cuestión. Por defecto son 2 para que éstas puedan ser mapeadas en un gráfico en dos dimensiones.
- *theta* es un parámetro que alude al algoritmo de Barnes-Hut (28), utilizado para aumentar la velocidad de computación del algoritmo. Es también un parámetro a tener en cuenta, ya que este incremento de la velocidad viene acompañado de una disminución en la precisión. Cuando su valor es 0, se utiliza una implementación estándar de t-SNE, por lo que cuanto más pequeño sea su valor, más exacta será la aproximación realizada por la función.

- *max_iter* es el número de iteraciones que debe realizar el algoritmo hasta encontrar una solución óptima; esto es, encontrar el mínimo de su función de coste.
- *momentum* es un término utilizado para reducir el número de iteraciones requeridas. Así, en lugar de guiarse por el gradiente actual, este parámetro acumula los gradientes de pasos anteriores para determinar la dirección a tomar por el algoritmo cuando realiza la búsqueda del mínimo de la función de coste (29).
- *final_momentum* tiene la misma utilidad que el parámetro anterior, pero éste es aplicado en las iteraciones finales, cuando los puntos del mapa están moderadamente bien organizados. Es aquí cuando el valor de este término debe ser más elevado, entrando en acción a partir de la iteración 250, de manera general.
- *exaggeration_factor* hace referencia al truco conocido como "*early exaggeration*". Éste consiste en multiplicar todas las probabilidades condicionadas de la alta dimensión por un escalar en las etapas iniciales de la optimización. De esta manera, se motiva a que esas grandes probabilidades sean modeladas mediante probabilidades también bastante elevadas en el espacio de baja dimensión, lo que hará que los grupos naturales estén muy separados en la representación.

4. Aplicaciones de t-SNE

Aunque la técnica t-SNE sea únicamente conocida como una técnica de visualización, ésta puede ser utilizada también para investigar o evaluar la segmentación de los distintos individuos o datos pertenecientes a un conjunto que conste de muchas características o dimensiones en diferentes grupos (o clústers).

En muchas ocasiones, el número de grupos o segmentos existentes en un conjunto de datos es seleccionado antes de modelar los resultados mediante un algoritmo de agrupamiento o clustering. El **análisis clúster o de conglomerados** básicamente se encarga de dividir el conjunto de datos en grupos significativos o útiles no conocidos de antemano. Este tipo de técnicas, catalogadas como de clasificación no supervisada, agrupan los objetos de datos basándose únicamente en la información contenida en los datos que describen a los objetos en cuestión y sus relaciones. El objetivo es que dichos objetos sean similares los unos a los otros dentro de cada grupo y que, a su vez, sean diferentes de los otros objetos pertenecientes a los otros grupos (30).

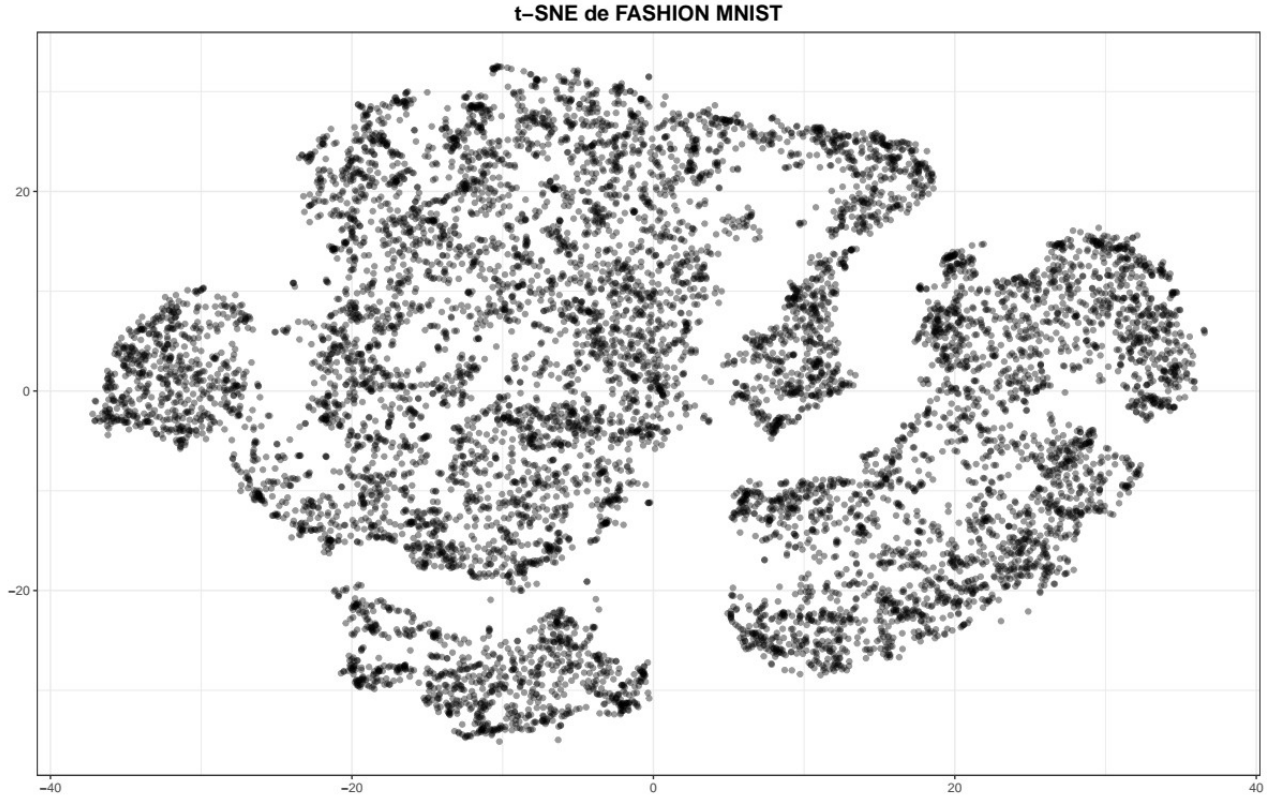
Como se ha podido apreciar en los ejemplos anteriores, t-SNE proporciona a menudo una representación en la baja dimensión en la que se puede observar una clara separación entre los datos. Entonces, ésta podría ser utilizada antes de usar cualquier algoritmo de clustering para seleccionar un número de grupos a priori, o incluso después para evaluar si éstos realmente se mantienen.

Sin embargo, se debe reseñar que t-SNE no es una técnica enfocada a la agrupación en sí misma, ya que, al contrario que otras técnicas tradicionales, no conserva las entradas y los valores a menudo pueden cambiar entre ejecuciones, por lo que solo debe ser usada para la exploración.

Para ejemplificar esta utilidad, se usará el conjunto de datos Fashion MNIST. Este conjunto corresponde nuevamente a 10.000 imágenes en escala de grises que tienen 784 píxeles y se corresponden a 10 categorías. Sin embargo, estas categorías corresponden a distintos artículos de moda, lo que lo hace muy interesante a la hora de de realizar agrupaciones entre los mismos.

Imaginemos una situación en la que no se conozcan las etiquetas de los datos, es decir, la clase de prenda a la que corresponde cada imagen. Un primer paso consistiría en representar en dos dimensiones este conjunto de imágenes mediante t-SNE, para así poder realizar una suposición a priori del número de grupos existentes en dicho conjunto:

Gráfico 19: t-SNE sobre Fashion MNIST. Datos no etiquetados.



En esta representación puede verse cómo los distintos artículos de moda forman algunos grupos bien definidos, aunque también existen otros que parecen estar más dispersos. Con todo, podría presumirse que las diferentes prendas se encuentran agrupadas en 4 o 5 grandes clases. Así, ese número de grupos establecido como suposición previa será el que se maneje a la hora de llevar a cabo cualquier método de clustering.

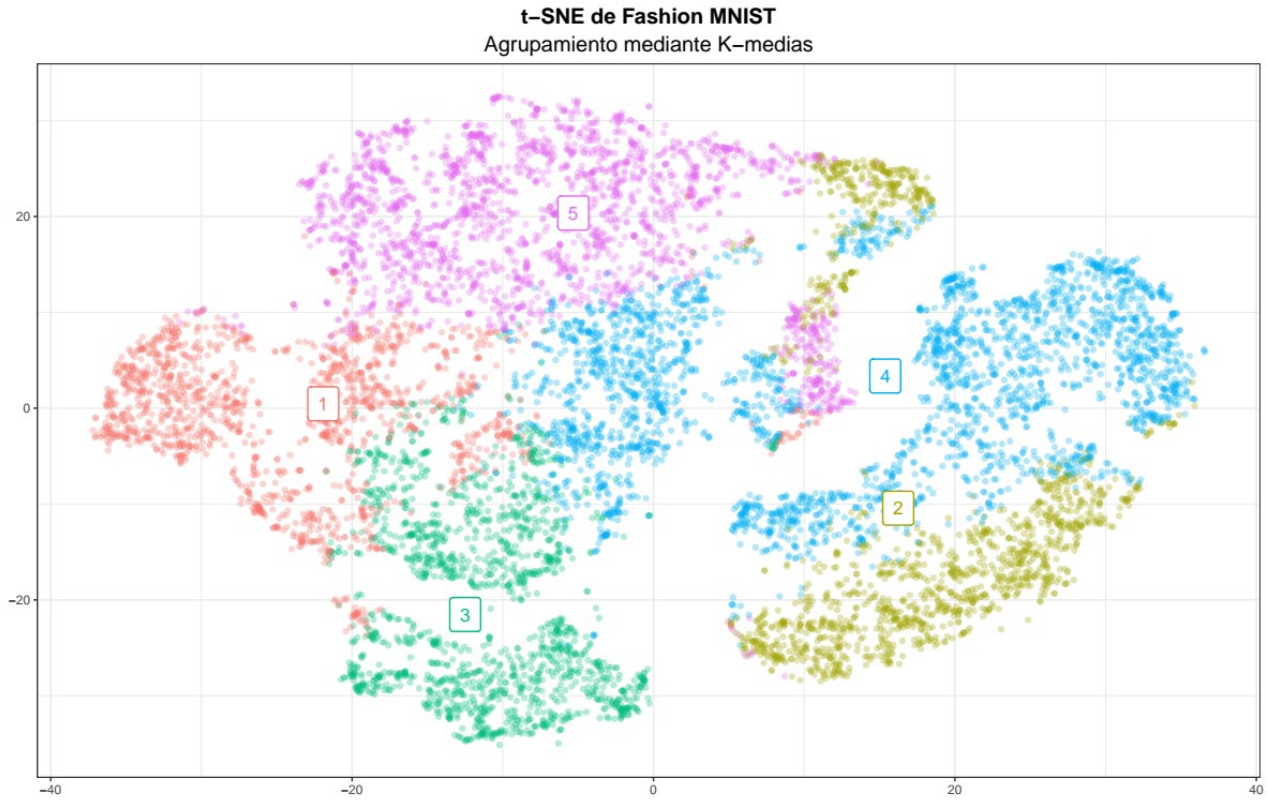
Para ver si dicha suposición es correcta, se aplicarán sobre los datos de Fashion MNIST dos de las técnicas de agrupamiento más conocidas y utilizadas: la técnica *K-medias* y el *agrupamiento jerárquico*.

K-medias es un método basado en prototipos. Esto significa que los clústers creados son conjuntos de objetos o individuos en los que cada uno de ellos es similar al objeto prototipo el cual define ese mismo grupo. En este caso, y como bien indica el nombre de la técnica, el individuo prototipo es definido como la media de todos los puntos en el clúster.

El funcionamiento de este algoritmo es sencillo, si bien es necesario conocer previamente cuantos grupos hay en la población. Entonces, ese número de grupos se corresponderá con los *K* centroides iniciales. Una vez especificado dicho parámetro, cada punto de los datos es asignado al centroide más cercano y cada conjunto de puntos asignados a un centroide formará un grupo. Puesto que el centroide de cada grupo se actualiza según se vayan asignando puntos al mismo, el proceso de asignación y actualización se repite hasta que ninguno de los puntos cambie de grupo o, de manera equivalente, hasta que los centroides sigan siendo los mismos.

Así, especificando como 5 el número de grupos a encontrar o establecer por el método de *K-medias*, se obtiene la siguiente partición de las distintas imágenes contenidas en el conjunto Fashion MNIST:

Gráfico 20: t-SNE sobre Fashion MNIST. Agrupamiento mediante K-medias.



Como bien se puede observar en el gráfico anterior en el que los artículos son etiquetados según el grupo establecido por *K-medias* al que pertenecen, no parece que éstos se correspondan con los bien separados clústers representados por t-SNE. Algunos de los grupos formados por la técnica se encuentran muy dispersos en la representación, como es el caso de los grupos 2 y 4, y también existen grandes regiones de puntos que son divididas en varios grupos. Sin embargo, este último caso quizás no sea tan preocupante, en el sentido de que se trata de particiones bastante densas, las cuales podrían encontrarse algo alejadas entre sí.

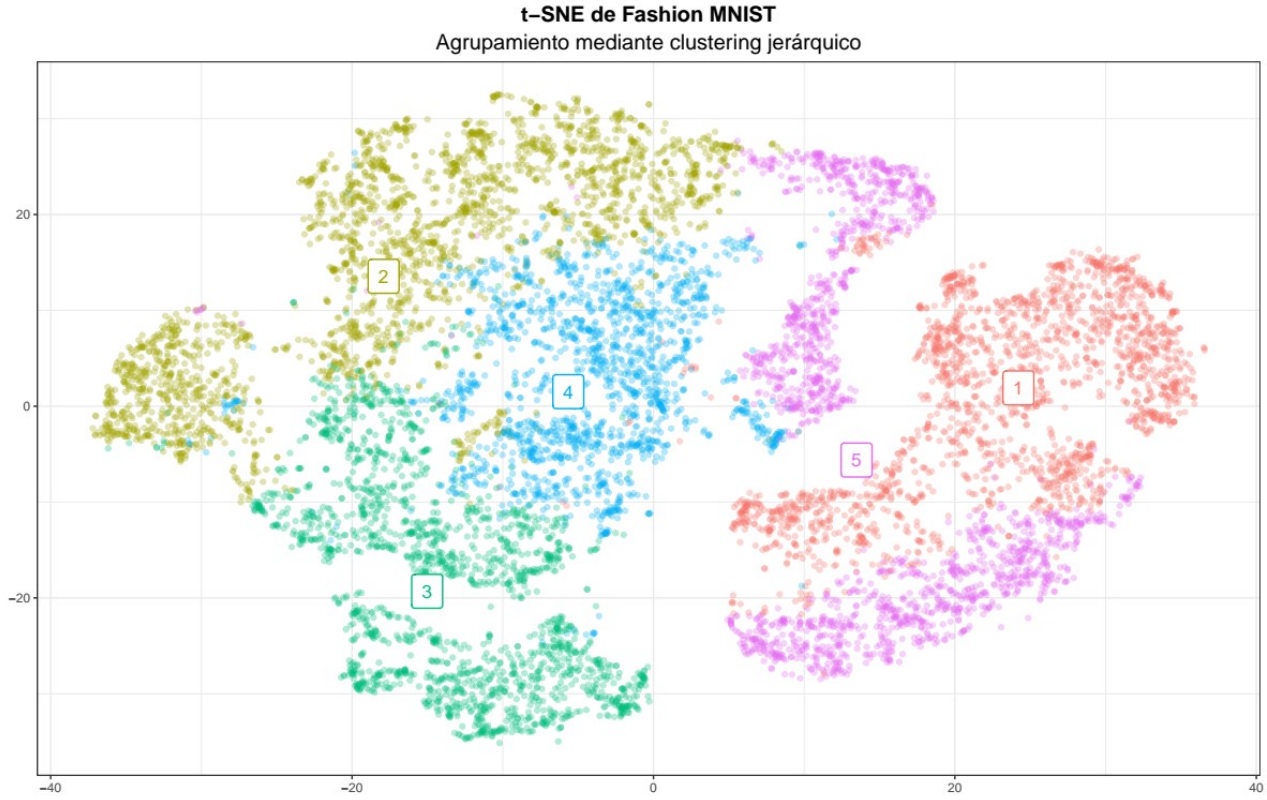
El segundo de los métodos de análisis clúster antes comentados es quizás el más común y conocido. El *agrupamiento o clustering jerárquico* es un enfoque cuyo algoritmo comienza tratando cada punto por individual como un clúster, por lo que parte de tantos grupos como individuos haya en el conjunto de datos. Después, se van realizando agrupaciones sucesivas entre aquellos grupos más parecidos, hasta llegar a un clúster final que contiene todos los casos analizados.

La clave de este algoritmo está en la computación de la proximidad entre los distintos clústers, y es la definición de esa proximidad o distancia lo que diferencia las diversas técnicas jerárquicas que pueden ser utilizadas. Para este caso concreto, se optará por escoger y aplicar el *método de Ward*, cuyo funcionamiento es similar al de *K-medias*, puesto que también asume que los grupos están representados por sus centroides. Sin embargo, este método mide la proximidad entre dos clústers

en términos del incremento en la suma de los cuadrados de los residuos (o SSE) resultante de su unión.

Cabe destacar que, a diferencia de *K-medias*, este método no necesita una especificación previa del número de grupos a formar. Más bien suele ser utilizado en casos en los que no se tiene una suposición concreta sobre el mismo, ya que el usuario podría escoger dicho número de grupos a posteriori, observando las proximidades o distancias entre éstos a través de, por ejemplo, la representación de un dendograma. No obstante, también es posible establecer previamente la cantidad de clústers deseados, como es este caso en el que dividimos el conjunto de imágenes en 5 grandes grupos:

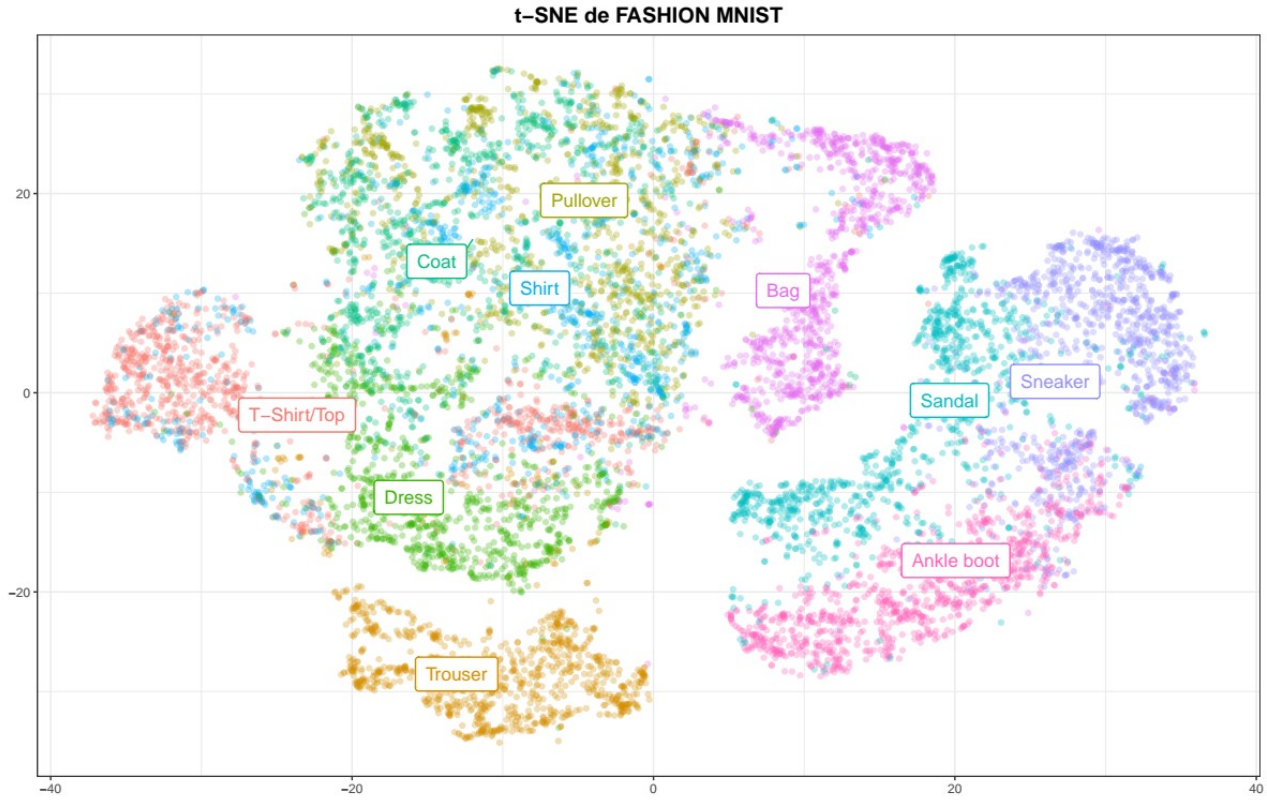
Gráfico 21: t-SNE sobre Fashion MNIST. Agrupamiento mediante clustering jerárquico.



De nuevo, se observa en la representación cómo los clústers formados por la técnica de clustering jerárquico tampoco se corresponden a los grupos de puntos que se pueden ver en el espacio, en muchos de los casos. Al igual que *K-medias*, los grandes grupos son divididos en varios subgrupos, pero al menos en este caso no se encuentran grupos tan dispersos en el espacio como en el anterior caso. Con todo, se trata de representaciones muy similares en cuanto a la partición del conjunto de datos.

Llegados este punto, resultaría interesante comprobar cual de los dos métodos realiza una mejor o más correcta agrupación de las diversas imágenes correspondientes a los distintos artículos de moda; es decir, si los clústers de puntos bien separados representados por t-SNE son grupos bien definidos por sí mismos, o si por el contrario, son aquéllos obtenidos a partir del análisis clúster los que verdaderamente hacen una buena distinción dentro de las distintas prendas de ropa. Para ello, es posible representar nuevamente el gráfico obtenido mediante t-SNE, al cual se añadirá la información relativa a la clase de artículo al que representa cada imagen:

Gráfico 22: t-SNE sobre Fashion MNIST. Datos etiquetados.



Como era de esperar, muchas de las diez clases totales en las que se divide el conjunto de imágenes se encuentran entremezcladas en la representación gráfica en dos dimensiones que se obtiene al aplicar t-SNE. Sin embargo, si se observa la agrupación que se realiza de las distintas etiquetas, la unión de esos grupos es totalmente plausible y aceptable.

El grupo de la derecha, el cual se encuentra bastante diferenciado del resto, está conformado por todas las imágenes correspondientes a todos aquellos artículos de calzado: sandalias (*sandal*), zapatillas deportivas (*sneaker*) y botas de tobillo (*ankle boot*). A su izquierda se encuentran agrupadas todas las imágenes de bolsos (*bag*), mientras que, en lo más abajo del gráfico, y también muy bien separado de los demás clústers, podemos encontrar el grupo que forman todas aquellas prendas correspondientes a pantalones (*trouser).* El clúster más grande es quizás el más

heterogéneo. En él encontramos tanto abrigos (*coat*) y suéteres (*pullover*), como camisas (*shirt*) y vestidos (*dress*). No obstante, todos esos artículos se corresponden a vestimentas que cubren la parte superior del cuerpo, por lo que sigue siendo un agrupamiento válido. El último grupo que podemos encontrar se encuentra muy próximo a éste y está compuesto por camisetas (*t-shirt*) y tops. Aunque parece raro que se encuentre algo distante del anterior, ya que también se corresponde a prendas de la parte superior, esta separación puede indicar alguna diferencia específica entre ellas, como por ejemplo las mangas, pues una característica de las camisetas es que suelen tener manga corta, a diferencia de los abrigos y suéteres.

Por su parte, las técnicas de análisis clúster que en este apartado fueron presentadas dividen esos grupos bien definidos, como bien se veía en las respectivas representaciones de las agrupaciones realizadas por *K-medias* (Gráfico 20) y el *agrupamiento jerárquico* (Gráfico 21).

Este resultado quizás es difícil de comprender, puesto que sería esperable que ambos métodos obtuvieran conclusiones muy parecidas sino similares, en el sentido en el que ambos utilizan y preservan las distancias en sus algoritmos. Quizás la principal diferencia se encuentre en el uso de la distribución t-Student por parte de t-SNE. Como ya se dijo, esta distribución es utilizada en favor de obtener una representación más fiel en el plano de las distancias moderadas existentes en la alta dimensión. Esto la hace captar una mayor estructura global de los datos, pero también puede hacer que la técnica pierda parte de la estructura local al realizar la representación, siendo de este modo distinguida de los métodos de clustering aquí presentados. Sin embargo, esta diferencia no está del todo clara, por lo que sería objeto de trabajos futuros el encontrarla.

5. Posibles problemáticas de t-SNE

Si bien en la sección principal en la que se hablaba sobre el método t-SNE ya se han comentado sus principales limitaciones e inconvenientes, existen algunos otros que pueden ser encontrados a la hora de utilizar esta técnica, pero que no descansan en el propio algoritmo de ésta. Esta última sección será aprovechada para comentar algunos de los problemas que el usuario podría encontrarse.

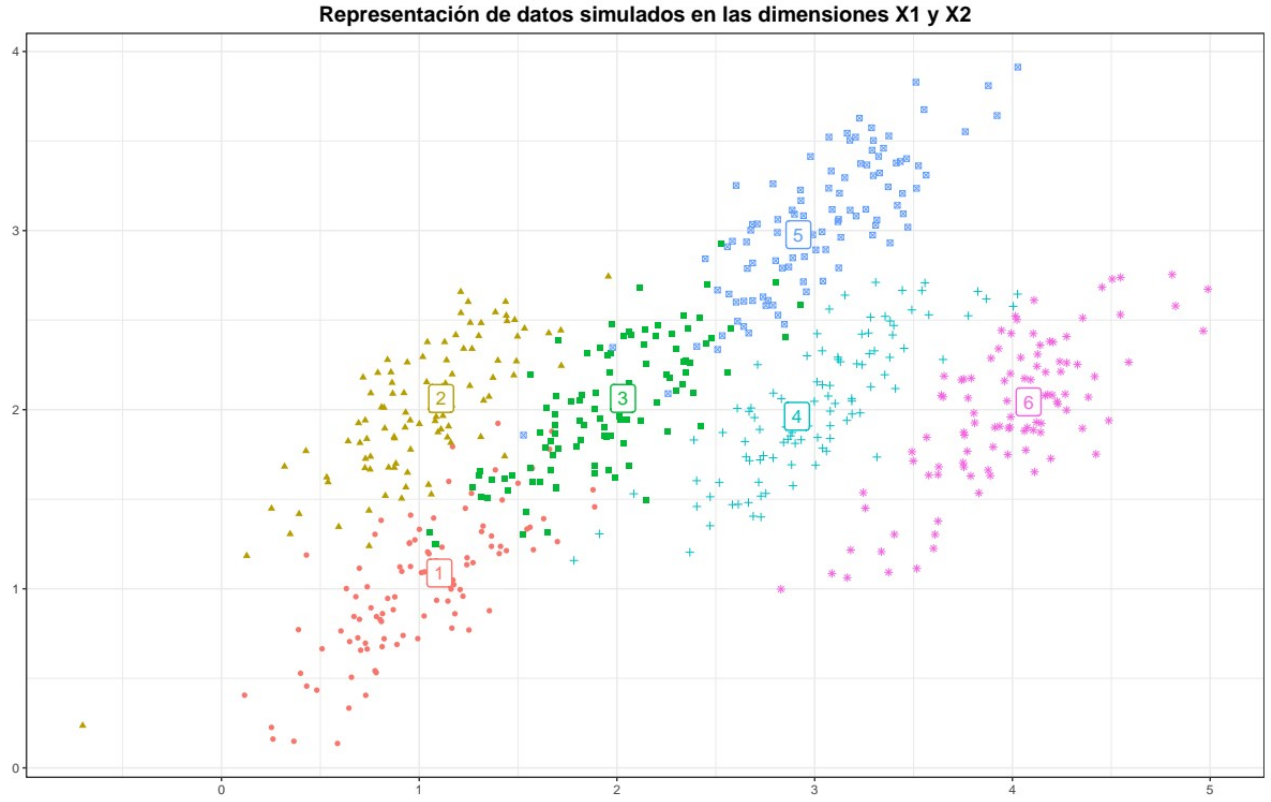
El primer problema es quizás el más usual de encontrar en este tipo de métodos. A lo largo de este documento, los ejemplos mostrados contenían datos referente a distintas imágenes, por lo que la magnitud de los datos era siempre la misma: píxeles. Para los casos en los que la técnica sea aplicada a conjuntos de datos en los que se tienen variables con diferentes unidades de medida, es preciso realizar una **normalización** o **estandarización** previa al uso de t-SNE. Esto significa

comprimir o extender los valores de las variables para que estén en un rango definido y se encuentren todas en la misma unidad de medida, permitiendo realizar comparaciones entre las mismas. Se trata de un paso muy importante en este caso, ya que, como se ha mostrado, t-SNE trabaja con la distancia Euclídea, la cual necesita trabajar con las mismas unidades de medida para que ninguna de las variables o dimensiones domine y tenga un peso mucho mayor en el cálculo de esta distancia.

Un segundo inconveniente podría surgir en el caso de seguir la recomendación del autor de t-SNE de aplicar previamente una reducción de la dimensión al conjunto de datos mediante PCA. Aunque es cierto que este procedimiento puede hacer reducir por mucho el coste computacional del algoritmo, también puede acarrear una pérdida de información importante y muy útil a la hora de separar los distintos grupos que pueden formar los datos.

Para ejemplificar este problema, pensemos en un ejemplo muy sencillo, en el que tenemos un conjunto de datos simulados con cuatro variables o dimensiones, en el que existen seis grupos bien separados por dos de esas variables, como se muestra en el siguiente gráfico:

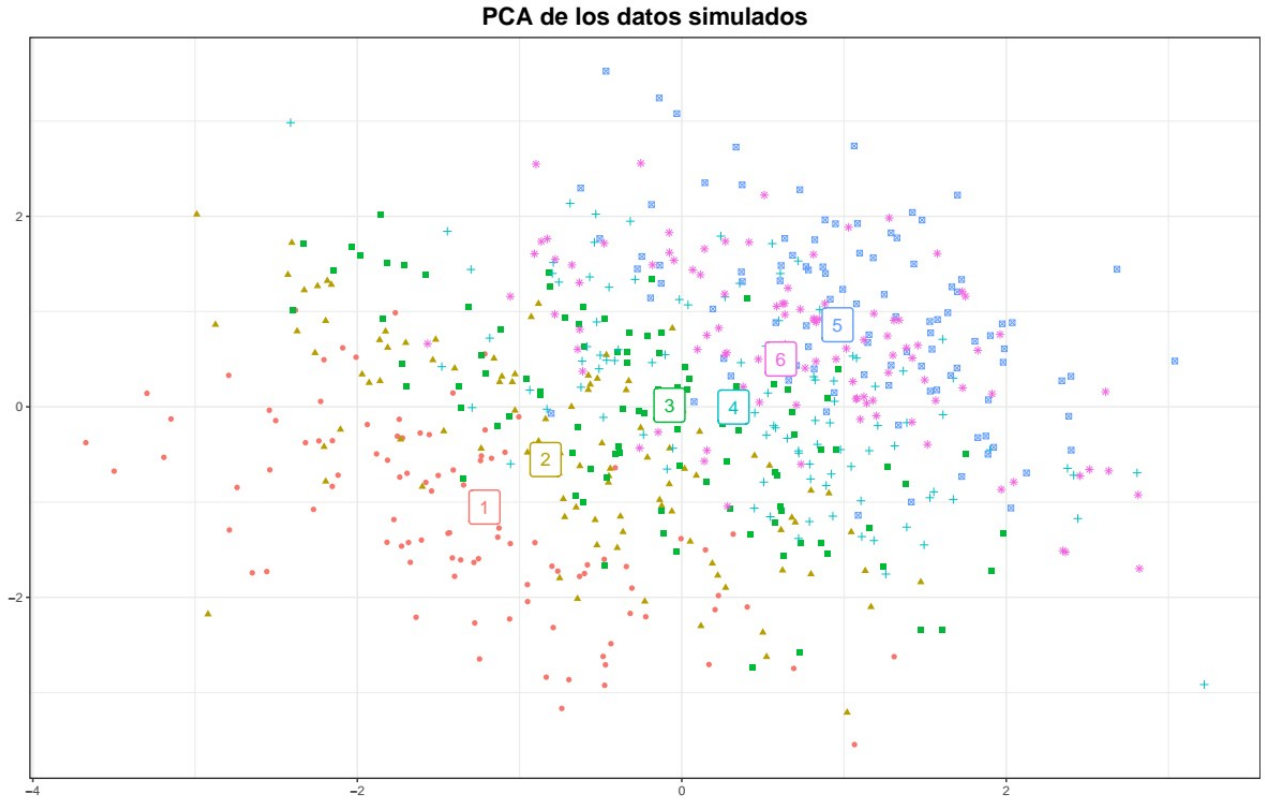
Gráfico 23: Representación de los datos simulados: dimensiones X1 y X2.



Puede verse como los grupos se encuentran muy bien definidos en el espacio en esas dos dimensiones. Ahora, imaginemos que se quisiera llevar a cabo una reducción de la dimensión del

conjunto total, es decir, de las cuatro dimensiones originales. Si se aplicara un PCA sobre este conjunto, se conseguiría una reducción del mismo a dos dimensiones, ya que por sí solas explican casi el 80% de la variabilidad contenida. No obstante, si proyectáramos los puntos sobre esas dos primeras componentes, obtendríamos un gráfico que tendría la siguiente forma:

Gráfico 24: Análisis de componentes principales de los datos simulados.



Ya no existe distinción alguna entre los seis diferentes grupos. Todos los puntos se encuentran ahora entremezclados, sin apenas posibilidad de distinguir entre ellos si pertenecen a uno o a otro grupo. Esto es debido a que, si bien las dos primeras componentes mantenían la mayor parte de la varianza de los datos, alguna de las otras que fueron descartadas debido a la poca información o variabilidad que capturaban era la que realmente realizaba la segmentación entre los clústers.

Este mismo problema podría ocurrir cuando se lleva a cabo una reducción de la dimensión previa a la ejecución de t-SNE. Es posible que al reducir y seleccionar las componentes a ser utilizadas por el algoritmo estemos dejando fuera aquélla que realmente muestra o establece la separación entre los distintos grupos a ser representados, en el caso en el que no explique o capture una parte significativa de la varianza de los datos, como bien se comento previamente.

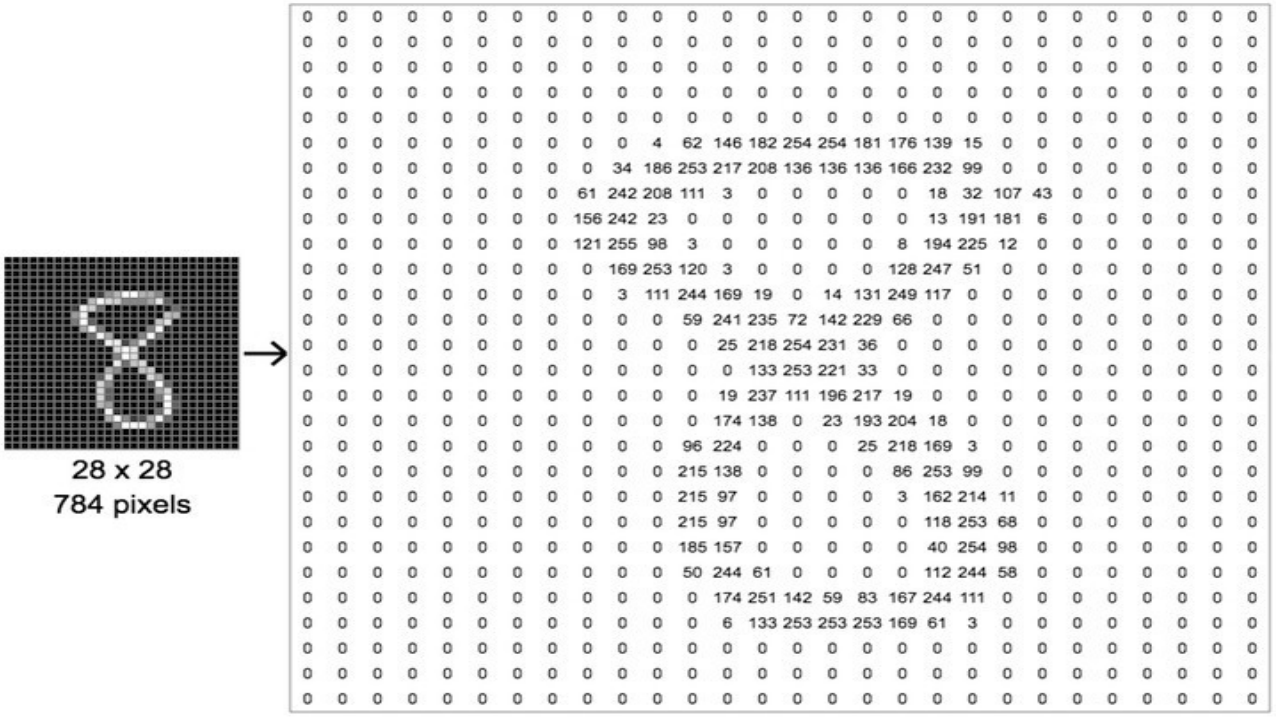
Una posible solución descansaría en sustituir este análisis de componentes principales previo por otro tipo de análisis como es el **clúster** antes mencionado. Este enfoque consistiría en seleccionar

una pequeña muestra de aquellos individuos que mejor agrupados y diferenciados estén dentro del conjunto. La clave sería obtener de esta forma muchos grupos muy diferentes entre sí, los cuales apenas reduzcan la variabilidad de cada variable. Una vez establecidos los distintos clústers, se podría aplicar un algoritmo de *K-medias*, en el que se eligiese como representante de cada uno de ellos un individuo o punto central. Por último, se irían representando el resto de puntos sobre el plano en base a la semejanza con uno u otro representante de cada grupo preestablecido. De esta manera, se garantizarían grupos muy homogéneos y se reduciría gran parte de la complejidad de la técnica, sin miedo a perder información importante contenida en el conjunto de datos.

El último de los inconvenientes a comentar en esta sección se encuentra también relacionado con el conjunto de datos al que se aplica esta técnica. Más concretamente, se encuentra relacionado con el tipo de objetos que tanto aquí como en la mayor parte de los trabajos referentes a t-SNE son representados en dos dimensiones: imágenes, que, además, en casi todos los casos se trata de imágenes correspondientes a números escritos a mano o alguna otra figura en escala de grises.

Como se ha comprobado en muchos de los apartados de este trabajo, t-SNE obtiene una muy buena representación y agrupación de los datos sobre el plano, sobre todo para el caso del conjunto de datos MNIST (Gráfico 6). La técnica logra separar muy bien las distintas imágenes según el dígito al que representan. Sin embargo, una característica que comparten todas ellas es que el número que contienen se encuentra muy centrado y, en la amplia mayoría, también muy recto, como bien se puede ver en la Ilustración 3:

Ilustración 3: Ejemplo de número escrito a mano del conjunto MNIST.

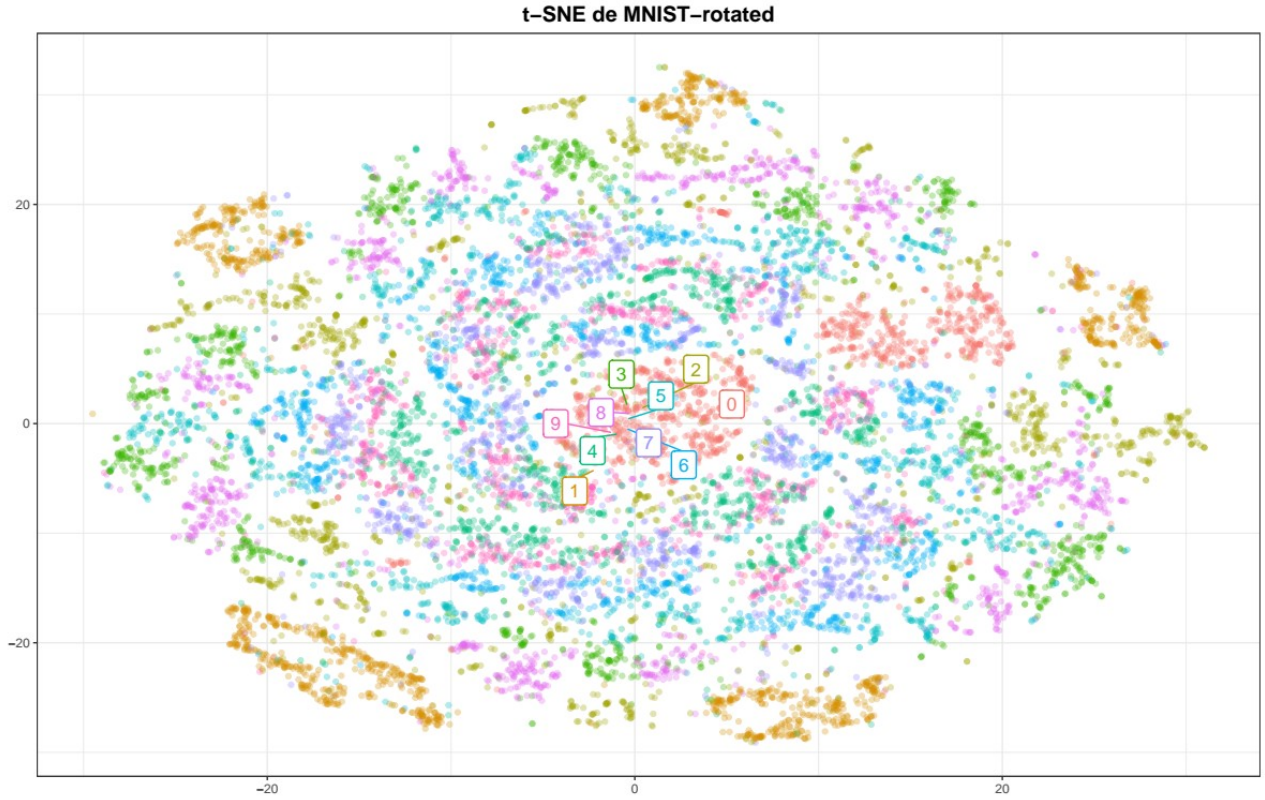


Fuente: https://towardsdatascience.com/how-to-teach-a-computer-to-see-with-convolutional-neural-networks-96c120827cd1

Este rasgo hace que pueda obtenerse una muy buena comparación entre las 784 componentes de las distintas imágenes píxel a píxel. Pero quizás este método no obtendría tan buenos resultados si se enfrentarse a unas imágenes no tan correctas de esos dígitos, ya sea por no estar dispuestas de manera tan centrada o por encontrarse rotadas en cierta medida.

Para comprobar este problema, aplicaremos t-SNE a una variante del conjunto MNIST. Este otro conjunto, conocido simplemente como MNIST-rotated, contiene las mismas imágenes que el anterior, con la salvedad de que éstas son rotadas en un ángulo generado uniformemente entre 0 y 2π radianes, lo que equivale a 360º. Representando gráficamente las coordenadas obtenidas por la técnica:

Gráfico 25: t-SNE sobre MNIST-rotated.



Claramente se ve como ésta no funciona de la misma manera. En este caso, en el que los dígitos no siguen un patrón tan claro y estricto, t-SNE no hace distinción alguna entre las diferentes imágenes. Tan solo se limita a representarlas gráficamente en círculos y de manera equidistante al centro, dónde se encuentra el grupo quizás mejor definido, aquél que contiene las imágenes correspondientes al número cero.

Este inconveniente no es fácil de resolver. Probablemente, para casos en los que se quiera trabajar con conjuntos de datos referentes a imágenes, sería más sencillo utilizar directamente algún otro método más contrastado y útil para este tipo de objetos, como el tan de moda **deep learning**. Por ejemplo, y más concretamente, podrían utilizarse las novedosas *redes neuronales convolucionales* (en inglés, CNN), las cuáles son capaces de coger una imagen, asignarle una importancia a varios aspectos/objetos contenidos en la misma y diferenciar una de otra (31). Con toda certeza, este algoritmo obtendría una mejor agrupación o clasificación de esos dígitos.

Para terminar, y muy unido al problema anterior, cabe destacar el hecho de que seguramente una representación de este tipo no será encontrada en ninguno de los trabajos o documentos referentes a t-SNE. En ellos únicamente encontraremos muy buenas representaciones de las diversas imágenes que, como se ha mostrado, están muy bien caracterizadas y son posibles de distinguir y agrupar entre ellas.

Ésto es algo conocido por los autores y demás usuarios profesionales de la técnica. La intención al aplicarla sobre este tipo de conjuntos de datos correspondientes a imágenes tan bien definidas es la de obtener una buena representación para mostrar al público lector. Sin embargo, este hecho haría que no se tratara propiamente de una técnica de análisis no supervisado, en el sentido en el que se conoce en gran medida el conjunto de datos al que se le aplicará, tanto en características y patrones como en posibles agrupaciones a formar. Así, podría ser visto más como un método de **análisis "presupervisado"**, utilizado para mostrar directamente los diferentes grupos existentes o contenidos en un conjunto ya conocido por el usuario. Es decir, t-SNE podría ser utilizada como herramienta de visualización a posteriori, una vez se haya realizado un análisis previo sobre los datos o se conozcan muy bien los objetos a representar, de forma que sepamos que obtendremos una representación correcta de los mismos.

6. Conclusiones

En este trabajo se ha estudiado detenidamente la técnica no lineal de análisis no supervisado de reducción de la dimensión t-Distributed Stochastic Neighbor Embedding (t-SNE), cuyo objetivo principal es la visualización de datos.

Tras comparar y comprobar cómo la técnica da lugar a mejores representaciones que en el caso de utilizar alguna otra técnica de reducción de la dimensión más tradicional, se analizó como afectaban los dos parámetros más importantes de su algoritmo a dicha representación:

- Establecer un valor bajo del parámetro *perplexity* puede llevar a representaciones donde los puntos de cada grupo se encuentren muy dispersos en el mapa. Si bien el incrementarlo hará que estos puntos se junten y definan de mejor manera los distintos grupos, establecer un valor demasiado exagerado o excesivo hará que los grupos lleguen a solaparse y no distinguirse entre sí.
- *Learning rate* (o tasa de aprendizaje) tiene diferentes efectos según el tamaño del conjunto de datos a representar. Para conjuntos no muy grandes, lo ideal sería establecer un valor no muy alto, puesto que valores muy elevados harían que los puntos se dispersaran en el mapa formando una especie de 'bola'. Por otro lado, la representación de conjuntos de datos muy grandes se ve beneficiada de elevadas tasas de aprendizaje, si bien también es posible establecer un valor bajo de ésta, siempre y cuanto se establezca un número de iteraciones suficiente para que el algoritmo de t-SNE pueda encontrar una solución óptima.

Además, se ha visto como la técnica en cuestión puede utilizarse para visualizar los diferentes grupos existentes y así establecer una suposición a priori del número de éstos. Aunque es cierto que dicha suposición no logra concordar con la solución que proporcionaría cualquier algoritmo de clustering, t-SNE consigue obtener una mejor o más correcta agrupación de los individuos a través de su representación. Sin embargo, no ha podido encontrarse una clara explicación al respecto, dejándose esta cuestión abierta para futuros trabajos relacionados.

Por último, se han mostrado los diferentes problemas que cualquier usuario de la técnica puede encontrarse al aplicarla, siendo causados la mayoría de ellos por los tipos de datos u objetos a representar. El estudio de estos inconvenientes ha permitido reflejar la importancia que recibe el conocimiento acerca del conjunto de datos a visualizar, lo que la convierte en una técnica de análisis "presupervisado" en lugar de no supervisado, puesto que su mayor utilidad surge cuando ya se tiene un buen conocimiento de los datos.

7. Bibliografía

(1) The Software Alliance (BSA). *¿Por qué son tan importantes los datos?* [en línea]. Washington, DC, 2015. Disponible en internet: https://data.bsa.org/wp-content/uploads/2015/10/BSADataStudy_es.pdf.

(2) López López, J.C. *La moda del Big Data: ¿en qué consiste en realidad?* [en línea]. Madrid, 25 de Febrero de 2014. Disponible en web: https://www.eleconomista.es/tecnologia/noticias/5578707/02/14/La-moda-del-Big-Data-En-que-consiste-en-realidad.html.

(3) Keim, D.A. Information Visualization and Visual Data Mining. *IEEE Transactions on visualization and computer graphics*, vol. 7, no. 1. pp 100-107, 2002.

(4) Izzati Kamsani, I., Abdul Khalid, N. E. y Ariff, N. Survey of Star Glyph-Based Visualization Technique of Multivariate Data. *Australian Journal of Basic and Applied Sciences*, vol. 9, no. 26, pp 61-66, 2015.

(5) Aftab, Z. y Tuaseef, H. Enhancing Piel Oriented Visualization by Merging Circle View and Circle Segments Visualization Techniques. *Multidisciplinary Trends in Artificial Intelligence*. Berlín: Springer-Verlag, 2012, pp 102-109.

(6) Shaffer, J.A. Using Treemaps to Visualize Data [en línea]. 4 de Marzo de 2017. Disponible en web: http://www.dataplusscience.com/UsingTreemaps.html.

(7) Orsenigo, C. y Vercellis, C.. Linear versus nonlinear dimensionality reduction for banks' credit rating prediction. *Knowledge-Based Systems*, no. 47, pp 14-22, 2013.

(8) Lever J., Krzywinski M. y Altman N. Principal Component Analysis. *Nature methods* [en línea], vol. 14, pp 641 – 642, 2017. Disponible en internet: https://www.nature.com/articles/nmeth.4346.

(9) Malhotra, V. Principal Component Analysis Deciphered. *Medium, SFU Big Data Science* [en línea], Marzo de 2015. Disponible en web: https://medium.com/sfu-big-data/principal-component-analysis-deciphered-79968b47d46c.

(10) *Multidimensional Scaling* [en línea]. NCSS Statistical Software. Disponible en internet: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Multidimensional_Scaling.pdf.

(11) Peña, D. Escalado Multidimensional. *Análisis multivariante*, pp 179 – 198.

(12) Chu, J. *Introducing Locally Linear Embedding (LLE) as a Mehod for Dimensionality Reduction*. San José State University, Department of Mathematics and Statistics, 2015.

(13) Roweis, S. T., Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, vol. 290, pp 2323 – 2326, 2000.

(14) Song, L, Smola, A. J., Borgdwardt, K. y Gretton. A. Colored Maximum Variance Unfolding. *Advances in Neural Information Processing Systems* [en línea], vol. 21, 2007. Disponible en internet: https://papers.nips.cc/paper/3307-colored-maximum-variance-unfolding.pdf.

(15) Weingberfer, K.Q., Packer, B. D., y Saul, L.K. Nonlinear Dimensionality Reduction by Semidefinite Programming and Kernel Matrix Factorization. *Advances in Neural Information Processing Systems* [en línea], vol. 19, 2007. Disponible en internet: http://www.cs.cornell.edu/~kilian/papers/171.pdf.

(16) Demartines, P. y Hérault, J. Curvilinear component analysis: A self-organization neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp 148 – 154, 1997.

(17) Hiton, G.E. y Roweis, S. T. Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems*, vol. 15, pp 833 – 840, 2002.

(18) Goyal, A. I do not understand t-SNE – Part 1. *Medium* [en línea], 1 de Enero de 2018. Disponible en web: https://medium.com/@layog/i-dont-understand-t-SNE-part-1-50f507acd4f9.

(19) *Kullback-Leibler Divergence* [en línea]. Popkes, A, Febrero de 2019. Disponible en internet: http://alpopkes.com/files/kl_divergence.pdf.

(20) Shannon, C.E. A mathematical theory of comunication. *The Bell System Technical Journal*, vol. 23, no. 3, 1948.

(21) *Algoritmo de la gradiente descendente.* Universidad TELESUP, 2017. Disponible en web: https://docplayer.es/73331882-Algoritmo-de-la-gradiente-descendente.html.

(22) Goyal, A. I do not understand t-SNE – Part 2. *Medium* [en línea], 21 de Enero de 2018. Disponible en web: https://medium.com/@layog/i-do-not-understand-t-SNE-part-2-b2f997d177e3.

(23) van der Maaten, L. y Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, vol. 9, pp 2579 – 2605, 2008.

(24) Cook, J. A., Sutskever, A. M. y Hinton G. Visualizing data with a mixture of maps. *Proceedings of the 11th International Conference on Articial Intelligence and Statistics*, vol. 2, pp 67 – 74, 2007.

(25) Wattenberg, M., Viégas, F. y Johnson I. How to Use t-SNE Effectively. *Distill* [en línea], 13 de Octubre de 2016. Disponible en web: https://distill.pub/2016/misread-tsne/#citation.

(26) Johnston, B., Jones, A. y Kruger, C. t-Distributed Stochastic Neighbor Embedding (t-SNE). *Applied Unsupervised Learning with Python*. Birmingham: Packt Publishing, pp 175 – 206, 2019.

(27) Kritjhe, J. *T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation (Rtsne)* [en software estadístico R]. Documentación disponible en internet: https://cran.r-project.org/web/packages/Rtsne/Rtsne.pdf.

(28) Ventimiglia, T. y Wayne, K. *The Barnes-Hut Algorithm* [en línea], 2011. Disponible en web: http://arborjs.org/docs/barnes-hut.

(29) Kathuria, A. *Intro to optimization in deep learning: Momentum, RMSProp and Adam* [en línea], 13 de Junio de 2018. Disponible en web: https://blog.paperspace.com/intro-to-optimization-momentum-rmsprop-adam/.

(30) Tan, P. N., Steinbach, M. y Kumar, V. *Cluster Analysis: Basic Concepts and Algorithms*, 2005.

(31) Saha, S. A Comprehensive Guide to Convutional Neural Networks – the ELI5 way. *Medium, Towards Data Science* [en línea], 15 de Diciembre de 2018. Disponible en web: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.