

Pontificia Universidad Católica de Chile
Exploratorio de Computación
29 de mayo de 2017

Informe Tarea Grande 2

Data Science

Integrantes: Wenyi He
Leonardo Olivares

(1) Tabla de algoritmos de clustering

Algoritmo		Nº de Clusters
• MeanShift		11
• DBScan	Default	0
	Eps=270, min_samples=3	8
	Eps=240, min_samples=2	15

(2) ¿Por qué se realiza la reduccion de 4096 a 50 dimensiones?

La razón por la cual se realiza la reducción inicial de 4096 dimensiones a 50 dimensiones, se debe a que es muy común que en un dataset (en este caso de vectores que representan imágenes), muchas de las columnas no aportan información útil para llevar a cabo el proceso de clustering, por lo que solo ralentizaría el tiempo de ejecución del algoritmo. Es por esto que PCA (el algoritmo utilizado para la reducción de dimensiones) tiene como objetivo determinar cuáles de las columnas ofrecen mejores datos, y a partir de estas, disminuir el dataset a uno que represente lo mismo pero que sea más eficaz de utilizar. El mecanismo utilizado por PCA para realizar la reducción es completamente matemático, y consiste principalmente en seleccionar las dimensiones cuyos datos tengan una mayor varianza. Una vez obtenido el dataset con 50 dimensiones, el proceso de clustering fue más eficaz y rápido que al no haberlo reducido.

(3) Tabla de algoritmos de clustering vs. Silhouette Score

Algoritmo		Silhouette_Score
• MeanShift		0.2178691
• DBScan	Default	N.A.
	Eps=270, min_samples=3	0.0946272
	Eps=240, min_samples=2	0.0513404
• K-Means	K = 10	0.0578405
	K = 20	0.0606830
	K = 30	0.0501364

(4) Elección de Algoritmo y Justificación

Al realizar la reducción de dimensiones, a un dataset con vectores de 50 dimensiones, se procedió a aplicar los distintos algoritmos de clustering requeridos y evaluar su eficacia con respecto a los demás. Para esto, se calculó el valor de `silouehette_score` para cada uno. Esta herramienta es un índice de validación, cuyo valor puede estar entre -1 y 1, siendo -1 un resultado que indica que el resultado del algoritmo de clustering utilizado para un respectivo dataset fue incorrecto, y 1 la mejor agrupación de clusters posible. Se puede observar en la tabla (3) los resultados de esta métrica para cada algoritmo.

Una vez que se obtuvieron los valores del `silouehette_score`, se observó de manera gráfica como quedaban agrupadas las imágenes, y se concluyó que este instrumento (`s_score`) no era suficiente para determinar el mejor algoritmo de clustering para cada caso.

Primero, utilizando el algoritmo K-Means, se ingresaron como parámetros la cantidad de clusters que se deseaban ($K = 10, 20, 30$), y se observó que en general, el valor de `silouehette_score` eran muy similares (casi iguales). Luego, observando los gráficos de clusters, se pudo detallar como se formaban grupos, en los cuales los datos de cada uno eran cercanos y los grupos eran de tamaños similares.

Luego, se utilizó con el mismo dataset el algoritmo MeanShift, cuyo método de clasificación es distinto al de K-Means y se basa principalmente en colocar clusters en las posiciones medias de distintos grupos de datos. El `silouehette_score` en este caso fue mayor a los obtenidos en los del algoritmo anterior, sin embargo, observando la gráfica de los clusters, la agrupación de los datos que se obtuvo fue muy dispareja, siendo la mayor parte del dataset perteneciente a un mismo cluster, lo cual experimentalmente no ayuda a clasificar las imágenes.

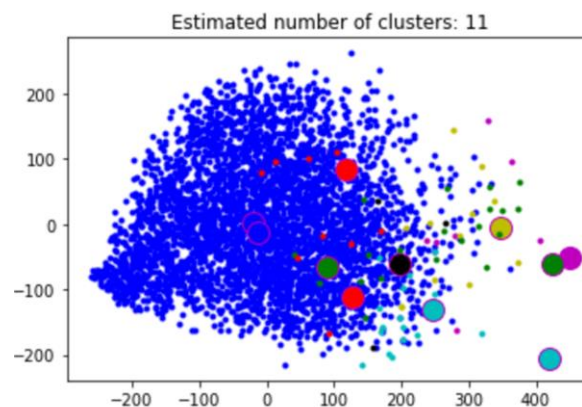


Gráfico de Clusters (MeanShift)

Por último, se aplicó el algoritmo DBScan, que como parámetros recibe un epsilon y min_samples, que se traduce a los cluster que tengan mínimo una cantidad de samples, dentro de una circunferencia de radio epsilon. Se observó la métrica de silhouette_score y arrojó resultados iguales o mas altos que el K-Means, por lo que se esperaba una mejor agrupación de las imágenes. Sin embargo, al observar un gráfico de los grupos obtenidos, se observa que para distintos valores de epsilon y min_samples, se generan clusters que no son ideales. Se pudo observar como se formaban clusters con muy pocos datos, mientras que la gran mayoría entraba en un mismo grupo. Además, muchas imágenes quedaban fuera de todos los clusters, lo cual aumentaba el error al clasificarlas.

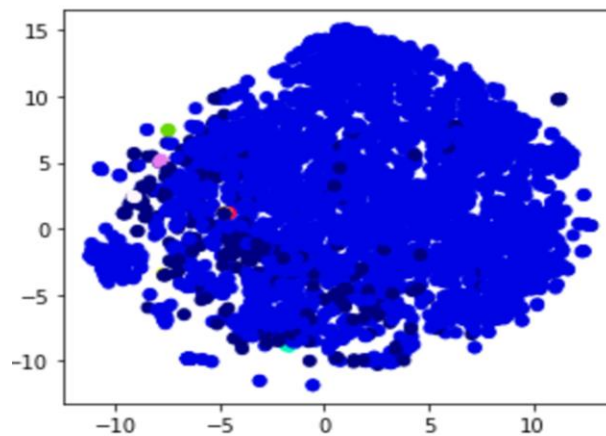


Grafico de Clusters (DBScan)

Por lo tanto, después de realizar este análisis detallado de cada algoritmo para el respectivo dataset, se seleccionó el algoritmo de K-Means utilizando como parametros $K = 30$, y la razón para esta decisión es la siguiente:

Se descartó el uso de Meanshift y DBScan, debido a que son procesos que trabajan con preferencia a los datos de mayor densidad, y en este caso nuestro dataset no se prestaba para su implementación, lo cual ocasionaba un balance incorrecto en los clusters, y una clasificación errónea. Entre los valores de K para K-Means se eligió el mayor valor (30), debido a que todos los silhouette_scores eran parecidos, y al tener mas clusters la clasificación de las imágenes terminaría siendo mejor y mas específica.

Con el metodo seleccionado, se realizo un nueva reduccion de dimensiones (de 50 a 2), con el fin de poder observar en 2 dimensiones las imágenes con los clusters obtenidos. El resultado del grafico de clusters de puede apreciar a continuacion, y muestra agrupaciones de los datos con sentido visual y con tamaños similares.

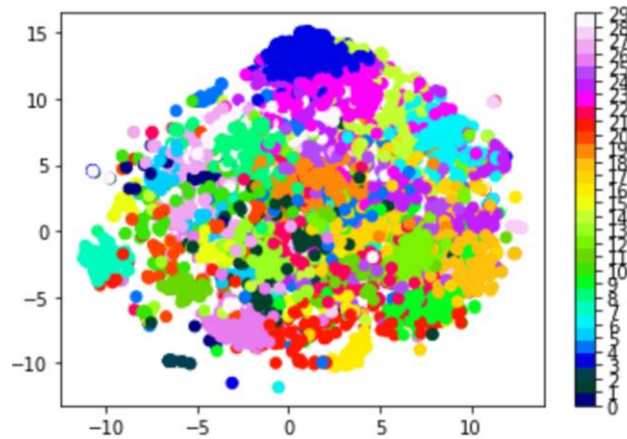





Grafico de Clusters (K-Means)

(5) Tipos de Características Clasificadas por K-Means (K=30)

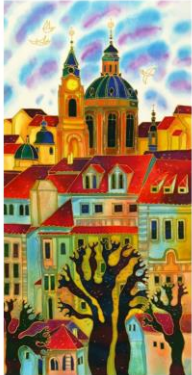


Los clusters que se obtuvieron con el algoritmo seleccionado (Kmeans, K = 30), clasificó las imágenes en grupos que , visualmente, tienen mucha concordancia, y por lo tanto se concluye que efectivamente, las imágenes se agruparon con otras según distintas características que las relaciona. A continuación se procedera a ejemplificar varios de los patrones utilizados por el algoritmo para asignar los clusters, analizando la visualización que genera la plantilla de HTML. Se aplicó el algoritmo de Jonker Volgenant, para observar de manera mas clara y eficiente las características de las imágenes.

Cluster 8

Detalle					Detalle					Detalle				
ID	Archivo	Clase	x	y	ID	Archivo	Clase	x	y	ID	Archivo	Clase	x	y
33870	byArtworkID/33870.jpg	Clase 8	320	341	34870	byArtworkID/34870.jpg	Clase 8	290	329	32419	byArtworkID/32419.jpg	Clase 8	286	318
														

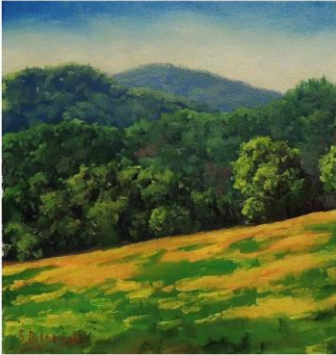

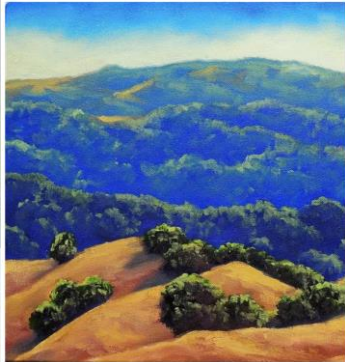
Se observan imágenes con figuras abstractas y uso de colores muy similares.

Cluster 13

Detalle					Detalle					Detalle				
ID	Archivo	Clase	x	y	ID	Archivo	Clase	x	y	ID	Archivo	Clase	x	y
31450	byArtworkID/31450.jpg	Clase 13	573	223	35609	byArtworkID/35609.jpg	Clase 13	572	221	30110	byArtworkID/30110.jpg	Clase 13	572	226
														

Se observan imágenes con edificios similares, y con usos de colores semejantes.

Cluster 15


Detalle					Detalle					Detalle				
ID	Archivo	Clase	x	y	ID	Archivo	Clase	x	y	ID	Archivo	Clase	x	y
31875	byArtworkID/31875.jpg	Clase 15	223	71	44360	byArtworkID/44360.jpg	Clase 15	238	87	28970	byArtworkID/28970.jpg	Clase 15	222	61
														

Se observan imágenes que muestran paisajes, principalmente montañas y áreas verdes.

Cluster 2


Detalle

ID	Archivo	Clase	x	y
35223	byArtworkID/35223.jpg	Clase 5	582	369




Detalle

ID	Archivo	Clase	x	y
30018	byArtworkID/30018.jpg	Clase 5	595	376



Detalle

ID	Archivo	Clase	x	y
39818	byArtworkID/39818.jpg	Clase 5	596	376




Se observan imágenes que muestran distintas frutas y diferentes tipos de jarrones, colocados sobre mesas.

Cluster 22


Detalle

ID	Archivo	Clase	x	y
39675	byArtworkID/39675.jpg	Clase 14	525	172




Detalle

ID	Archivo	Clase	x	y
38897	byArtworkID/38897.jpg	Clase 14	529	174



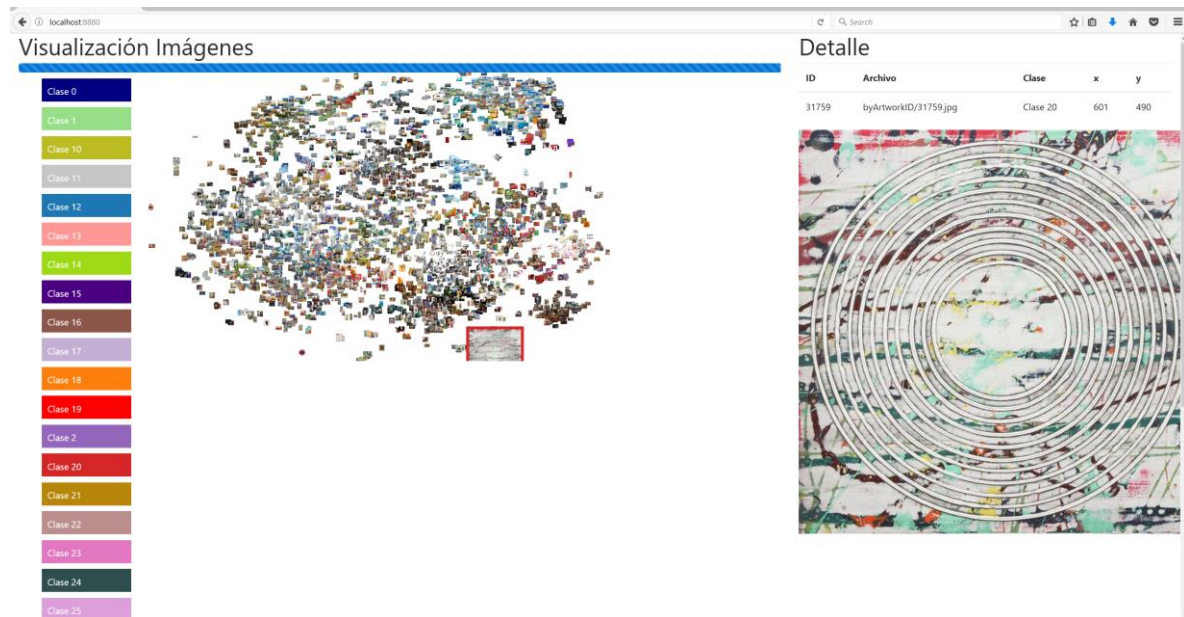
Detalle

ID	Archivo	Clase	x	y
26269	byArtworkID/26269.jpg	Clase 14	534	182



Se observan imágenes con curvas que asimilan a la ola del mar, también presentan colores de temperatura similar.

Visualización



Utilizando la plantilla HTML y JS del ayudante, se visualiza las imágenes ordenado por T-SNE y con los clústeres de K-Means implementado. Solo para el motivo de poder ver los nombres de los clústeres en la izquierda, se modifica el código de JS para que los botones no se quedan con fondo blanco. A la línea 10 del viz.js, se cambia a:

```
var tableau20 = [  
  '#1f77b4', '#aec7e8', '#ff7f0e', '#ffbb78', '#2ca02c',  
  '#98df8a', '#d62728', '#ff9896', '#9467bd', '#c5b0d5',  
  '#8c564b', '#c49c94', '#e377c2', '#7b6d2', '#7f7f7f',  
  '#c7c7c7', '#bcbd22', '#dbdb8d', '#17becf', '#9eda16',  
  '#800000', '#000080', '#2F4F4F', '#808000', '#4B0082',  
  '#B8860B', '#FF0000', '#DDA0DD', '#778899', '#BC8F8F'];
```

Así cada botón tendrá un color de fondo para poder verlos.

LAPJV

← localhost:8080

Visualización Imágenes

Clase 0

Clase 1

Clase 10

Clase 11

Clase 12

Clase 13

Clase 14

Clase 15

Clase 16

Clase 17

Clase 18

Clase 19

Clase 2

Clase 20


Clase 21

Clase 22

Clase 23


Clase 24

Clase 25



Detalle

ID	Archivo	Clase	x	y
35790	byArtworkID/35790.jpg	Clase 18	787	285



Utilizando el csv: “data_lapjv.csv” incluido en el repo, se visualiza la implementación de LAPJV en la tarea. Se muestra los primeros 4096 imágenes en el grid de 64*64. Se puede notar que el algoritmo de clasificación funciona, y se nota que las imágenes similares se quedan relativamente cercanos por su parte.