Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
Computación: Ciencia y Tecnología del Mundo Digital

Tarea Grande 2 - Data Science

IIC1005 - 1er semestre de 2017
Profesor Denis Parra
3 de mayo de 2017

Indicaciones

- Fecha de entrega: 29 de mayo de 2017.
- Debes entregar la tarea en tu repositorio privado de GitHub.
- Cada hora de atraso descuenta 5 décimas de la nota que obtengas.
- Esta tarea es en grupos de máximo 2 alumnos. La copia será sancionada con una nota 1.1 en la tarea, además de las sanciones disciplinarias correspondientes.
- Debes indicar a más tardar el 11 de mayo de 2017 vía cuestionario en el siding quiénes conforman tu grupo y el repositorio en el donde se subirá la tarea.

Objetivos

El objetivo de esta tarea es que aprendas a:

- Utilizar Python para trabajar con librerías dedicadas al aprendizaje de máquina (*sk-learn*)
- Utilizar algoritmos de reducción de dimensionalidad.
- Utilizar javascript para finalmente realizar una visualización de los datos y corroborar visualmente aglomeraciones o patrones en los datos.

Introducción

En nuestra vida como científicos los datos son esenciales. Gracias a ellos se han logrado grandes avances, como el reconocimiento facial o la detección de anomalías en los exámenes médicos. Ahora, cuando tenemos datos en 2 dimensiones es fácil generar una visualización. En 3 dimensiones también es posible, incluso se puede hasta 4 (utilizando la cuarta dimensión como el color de los puntos por ejemplo, variando el tono o el brillo del color nos puede dar una cuarta dimensión a visualizar). Pero, ¿qué podemos hacer si tenemos datos en 2000 o 3000 dimensiones?. Para su sorpresa, en esta tarea trabajarán con datos en **4096** dimensiones. Más sorpresivo será que, si logran realizarla completa, podrán generar una visualización en 2 dimensiones que a la simple vista les permitirá ver patrones o aglomeraciones en los datos.

Instrucciones

Para esta tarea se te facilitará un archivo comprimido (zip) que contiene una carpeta donde están 4214 imágenes de obras de arte etiquetadas con un *id*, en formato *id.jpg*. Además contiene un archivo de texto que en cada linea contiene *nombrearchivo:descriptor*. Cada descriptor es un vector de 4096 dimensiones que deberán manipular según los pasos a continuación para lograr el objetivo de esta tarea y luego escribir un reporte sobre los resultados.

Paso a paso

En la librería *sk-learn* de Python están implementados todos los algoritmos necesarios que deben utilizar.

1. (0,5 puntos) Utilizar el procedimiento de reducción dimensional PCA (Principal Component Analisys) para convertir cada vector (imagen) de **4096** dimensiones a un nuevo vector de **50** dimensiones.
2. Deberán generar clusters con los vectores de 50 dimensiones con distintos algoritmos.
   - (0,5 puntos) K-Means, con $K = \{10, 20, 30\}$.
   - (0,5 puntos) Mean Shift
   - (0,5 puntos) DBSCAN, deberán utilizar 3 combinaciones de parámetros distintas para *epsilon* y el mínimo de puntos que deben pertenecer a una región (utilice los parámetros por defecto y otras dos combinaciones arbitrarias que elija).
3. (0,5 puntos) Una vez que tengas los clusters listos, tendrán que realizar una nueva reducción de dimensionalidad. Sin embargo, en este paso, no utilizarás los clusters del punto 2: esta vez, de las 50 dimensiones del punto 1, deberán reducirlas a 2 dimensiones, con el proposito de poder visualizar los clusters de imágenes. El algoritmo de reducción a utilizar será *t-SNE* (t-distributed Sthocastic Neighbor Embedding), este algoritmo es mucho más efectivo para reducir a 2 o 3 dimensiones, pero te recomendamos, para esta tarea fijar el parámetro **perplexity** = 50.
4. (0,5 puntos) Por último, deberán generar un archivo csv para para utilizar la visualización implementada en JavaScript (esta te la entrega implementada el ayudantes) con el siguiente formato:

id	x	y	*cluster*
43526	3.524	1.2645	7
⋮	⋮	⋮	⋮

Entregables

El código y el informe de esta tarea deben ser entregados en el repositorio privado de GitHub del integrante indicado en el cuestionario de siding.
Código (3 puntos): Deberán entregar la tarea en un Jupyter Notebook.
Visualización interactiva en HTML+JS (1 punto): Usando los datos del punto 4 y la plantilla de HTML+JS generada por los ayudantes, entregar la página HTML que muestre las imágenes de forma interactiva.
Informe (2 puntos): Escribir un informe en PDF con el siguiente contenido.

1. (0,3 puntos) Tabla que indique la cantidad de clusters generados por cada algoritmo según los parámetros utilizados (mean shift y dbscan, ya que para K-means el numero de clusters es un parámetro de entrada)
2. (0,2 puntos) ¿Por qué se realiza la reducción de 4096 a 50 dimensiones en el primer paso?

3. (0,5 puntos) Gráfico o tabla que muestre la métrica *silhouette_score* vs los parámetros elegidos por cada algoritmo de clustering (se recomienda investigar sobre la interpretación de esta métrica).

4. (0,5 puntos) Determinar qué algoritmo y con qué parámetros se entregan los mejores resultados y argumentar el porqué de la decisión.

5. (0,5 puntos) Del algoritmo y parámetros que determinó en el punto anterior, revise los clusters generados y argumente qué tipo de características cree usted fueron clasificadas. Agregue screenshots de la visualización interactiva para explicar.

Bonus

(1 punto) Esta tarea fue demasiado fácil y necesitas un verdadero desafío.
Puedes optar a **un punto** extra si aplicas el algoritmo *Jonker-Volgenant* para colocar las imágenes en un grid de $64 * 64$ casillas, tomando las primeras 4096 imágenes. Para orientarte sobre lo que debes realizar está el siguiente link