Trabajo Final - Informe II: Propuesta de Trabajo

Cipolatti Edgardo, Rosales Mario y Santellán Franco
edgardocipolatti@hotmail.com - mariorosales941@gmail.com - fransantellan@gmail.com

I. Introducción

A lo largo del curso se han desarrollado algoritmos para la implementación de diferentes redes neuronales, siempre siguiendo como objetivo común la resolución de problemas. A la hora de clasificar se implementó, entre otras, una red neuronal con funciones de base radial que nos permitió identificar patrones entre varias clases. Dicha red se basa en funciones *gaussianas* creadas a partir de dos parámetros, la *media* μ y la *varianza* σ. Para hallar μ, se implementó el algoritmo *k-means clustering standard*, que encuentra k centroides según la dispersión y agrupamiento de los patrones, para luego ser usados como parámetro μ.

Al implementar *k-means clustering standard*, nos encontramos con el problema de que la convergencia de dicho algoritmo dependía totalmente de la inicialización de los primeros centroides, denominados *semillas*. Donde, ante una mala inicialización, se encontraban centroides no convenientes que comprometían el desempeño de la red y, posteriormente, la clasificación.

Por lo dicho anteriormente, nos vimos intrigados en la búsqueda de otros métodos de inicialización de semillas que no sea una simple inicialización al azar.

II. Métodos de Inicialización a Analizar

En la búsqueda para dar con un conjunto de *semillas* que sea una buena aproximación a centroides, proponemos analizar los siguientes métodos (ver [1]–[3]):

- McQueen (1967)
- Etiquetado
- McRae (1971)
- Forgy (1965)
- Astrahan (1970)
- k-means++ (2007)

Suponiendo que el número de clusters a formar es k, entonces:

A. McQueen (1967)

El método de McQueen se basa en elegir como semillas a los k primeros patrones del conjunto de datos. En este caso se debe tener en cuenta que es necesario mezclar los datos para eliminar la dependencia que puedan llegar a tener entre ellos.

B. Etiquetado

Se etiquetan los patrones de 1 a m y se eligen como semillas aquellos patrones cuya etiqueta se obtiene de la siguiente manera:

$$\left[\frac{\alpha \cdot m}{k}\right]$$

donde $\alpha = 1, 2, \ldots, (k-1), k$, y donde $[x]$ representa la parte entera de x.

Con este sistema se pretende compensar la tendencia natural de ordenar los casos en el orden de introducción o alguna otra secuencia no aleatoria.

C. McRae (1971)

Se etiquetan los patrones de 1 a m. Para obtener la primer semilla, se genera un número al azar entre 1 y m, dicho número indica el patrón seleccionado. Para la siguiente semilla, se repite el mismo procedimiento pero esta vez, generando un valor al azar entre 1 y $(m-1)$, debido a que ya se ha obtenido una semilla y la cantidad de patrones disminuye una unidad.

D. *Forgy (1965)*

Para éste método lo que se hace es formar k grupos de patrones mutuamente excluyentes y usar sus centroides como semillas.

E. *Astrahan (1970)*

El algoritmo de Astrahan, el cual nos permite elegir las semillas de tal forma que abarquen todo el conjunto de datos, es decir, los datos estarán relativamente próximos a un punto semilla, pero las semillas estarán dispersas unas de otras. Astrahan propuso el siguiente algoritmo para ello:

- Para cada individuo se calcula la densidad, entendiendo por tal el número de casos que distan de él una cierta distancia, digamos d_1.
- Ordenar los casos por densidades y elegir aquel que tenga la mayor densidad como primer punto semilla.
- Elegir de forma sucesiva los puntos semilla en orden de densidad decreciente sujeto a que cada nueva semilla tenga al menos una distancia mínima, d_2, con los otros puntos elegidos anteriormente. Continuar eligiendo semillas hasta que todos los casos que faltan tengan densidad cero, o sea, hay al menos una distancia d_1 de cada punto a otro.

F. *k-means++*

Este algoritmo es propuesto por David Arturo y Sergei Vassilvitskii en 2007. *k-means++* consiste en encontrar semillas tal que se minimice la varianza entre grupos, es decir, minimizar la suma de las distancias al cuadrado de cada punto al centro mas cercano a él.

Sean X el conjunto de patrones y $D(x)$ la distancia mínima desde un patrón x al centro mas cercano. Entonces, el algoritmo *k-means++* se define como:

- Se toma un centro c_1, elegido uniformemente al azar de X.
- Tomo un nuevo centro c_i, eligiendo $x \in X$ con probabilidad:

$$P = \frac{D(x)^2}{\sum_{x \in X} D(x)^2}$$

- Repetir el paso anterior hasta que hallamos tomado k centros.
- Se procede al algoritmo *k-means clustering standard* con las k semillas encontradas.

III. Aplicación y Evaluación de los Métodos

A los efectos de establecer la calidad de los clusters encontrados por el algoritmo *k-means clustering standard*, usando la inicialización de los distintos algoritmos propuestos, proponemos 4 medidas posibles (ver [4], [5]):

- **Distancia Intra-Cluster** (cohesión). Distancia entre todos los elementos de un cluster, el objetivo es minimizar esta distancia:

$$\frac{\sum_{i=1}^{K} (\sum_{z,t \in C_i} d(z,t) / \, |C_i|)}{K}$$

- **Distancia Inter-Cluster** (separación). Distancia entre los centroides de los clusters, donde el objetivo es maximizar la distancia entre ellos:

$$\frac{\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} d(c_i, c_j)}{\sum_{i=1}^{K-1} i}$$

- **Pureza**. A cada cluster se le asigna una clase, la cual es la más frecuente en el cluster, y la medida se realiza contando la cantidad de elementos que pertenecen a esa clase dentro del cluster dividido la totalidad de elementos en el cluster. Formalmente:

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j|.$$

- **Matriz de confusión o tabla de contingencia**. A partir de la información brindada por la misma podemos extraer indices como el RI (Rand index), F score entre otras.

Por otro lado, para tener otras medidas de cuan buenos son los algoritmos utilizados, se propone analizar el rendimiento del algoritmo *k-means clustering standard*. Utilizando cada una de las inicializaciones proporcionadas por los algoritmos propuestos, se consideran los siguientes aspectos:

- Tiempo de ejecución
- Cantidad de Iteraciones

Todos estos métodos serán evaluados utilizando las siguientes bases de datos:

- *Iris Plants*: es una base de datos con 4 atributos numéricos, 3 clases y 150 elementos. La misma fue proporcionada por la cátedra.
- *Wine Quality*: una base de datos para clasificar vinos según su calidad, obtenida de *UCI Machine Learning Repository*. La misma se divide en casos casos para *Vino Blanco* y *Vino Tinto*, con 4899 y 1600 casos respectivamente. Estas casos contienen 12 atributos de los cuales los primeros 11 son características fisico-químicas, mientras que el último atributo representa la calidad exacta del vino.
- *Glass Identification*: también obtenida de *UCI Machine Learning Repository* y contiene 214 casos con 10 atributos cada uno. Ésta base de datos contiene 6 tipos de vidrio definidos en términos de su contenido de óxido (es decir Na, Fe, K, etc).

REFERENCIAS

[1] P. Larrañaga, I. Inza, y A. Moujahid, *Tema 14. Clustering*. Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad del País Vasco, 2014.

[2] "Métodos no jerárquicos de ana?lisis cluster." 2014, capítulo 4.

[3] D. Arthur y S. Vassilvitskii, *k-means++: The Advantages of Careful Seeding*. Stanford Theory Groups, 2007.

[4] A. Villagra, A. Guzmán, D. Pandolfi, y G. Leguizamo?n, *Ana?lisis de medidas no-supervisadas de calidad en clusters obtenidos por K-means*. Universidad Nacional de la Patagonia Austral and Universidad Nacional de San Luis, 2012.

[5] S. Schulte im Walde, "Experiments on the Automatic Induction of German Semantic Verb Classes," Disertación doctoral, Institut for Maschinelle Sprachverarbeitung, University Stuttgart, 2003, published as AIMS Report 9(2).

[6] Y. Xu, W. Qu, Z. Li, G. Min, K. Li, y Z. Liu, "Efficient k -means++ approximation with mapreduce," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3135–3144, Dec 2014.

[7] U. Maulik y S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, Dec 2002.