Mejora del algoritmo K-means mediante una meta-heurística orientada a la reducción de su complejidad computacional

Joaquín Pérez, Miguel Hidalgo,
Nelva Almanza, Noé Castro, Vitervo López
CENIDET
Morelos, México
Email: jpo_cenidet@yahoo.com.mx

Hugo Estrada
INFOTEC
México

Adriana Mexicano
Instituto Tecnológico de Cd. Victoria
Tampico, México
Email: mexicanoa@gmail.com

***Abstract*—El algoritmo de agrupamiento K-means ha tenido un uso generalizado en varios dominios y en una gran cantidad de aplicaciones, debido a su facilidad de implementación computacional, sin embargo, una de sus limitaciones es su alta complejidad computacional. En este sentido se propone una nueva meta-heurística, a la que denominamos N-means, que permite reducir la complejidad de K-means de manera importante, posibilitando que con los mismos recursos computacionales se resuelvan instancias más grandes y en menor tiempo. Para validar la propuesta, se realizaron experimentos con datasets reconocidos por la comunidad científica, y se contrastaron los resultados con los obtenidos con el algoritmo K-means estándar. En particular se encontró que con N-Means se obtuvieron reducciones de tiempo hasta de un 91% y una disminución de la calidad de sólo 5.5%. Es destacable que con base en el análisis de los resultados se observó un comportamiento cuasi lineal de N-means.**

I. Introducción

Actualmente en diferentes áreas del conocimiento se presenta un interés por entender los datos u objetos que explican o describen hechos particulares de un dominio. En ese sentido, una de las principales técnicas para conseguir la obtención de ese conocimiento es el agrupamiento de objetos.

El agrupamiento consiste en dividir un conjunto de objetos en subconjuntos, de tal manera que los objetos de un subconjunto son similares entre sí y disimilares de los objetos de los otros subconjuntos [1].

Uno de los algoritmos más populares para realizar el agrupamiento de datos es K-means [2], también conocido como el algoritmo de Lloyd [3]. Este algoritmo se caracteriza por su simpleza y fácil implementación. Sin embargo, su complejidad computacional es alta, lo cual limita su uso en instancias de gran tamaño. Adicionalmente algunos autores han reconocido que el problema de agrupamiento es de tipo NP [4],[5],[6].

Los pasos principales del algoritmo K-means estándar se presentan a continuación:

1) Inicialización:
   Consiste en definir los objetos que se van a particionar, el número de grupos y el centroide para cada grupo. Se han propuesto varios métodos para definir los centroides iniciales [7], [8], [9], [10], [11], [12], [13] y [14], aunque el enfoque más utilizado sigue siendo la selección aleatoria.
2) Clasificación:
   Para cada objeto se calcula su distancia hacia todos los centroides; se identifica el centroide más cercano y se asigna el objeto al grupo asociado con dicho centroide.
3) Cálculo de centroides:
   En este paso se recalcula el centroide para cada grupo generado en el paso anterior.
4) Criterio de convergencia:
   Consiste en establecer el criterio de paro del algoritmo, por ejemplo: cuando se alcanza un número determinado de iteraciones; cuando no hay intercambio de objetos entre los grupos o cuando la diferencia de los centroides en dos iteraciones consecutivas es menor a un umbral dado. Si no se satisface la condición de convergencia se repiten los pasos 2, 3 y 4.

En este artículo se propone una mejora de K-means con objeto de reducir su alta complejidad computacional, mediante la reducción del número de cálculos de distancia de los objetos a los centroides. A continuación se describen otros trabajos orientados a obtener una reducción de la complejidad de K-means.

El algoritmo propuesto en [15] reduce el número de cálculos de distancia de un objeto a los centroides porque almacena para cada objeto la distancia al centroide más cercano o asignado, en cada iteración. De este modo, en cada iteración se calcula la distancia del objeto hacia el centroide asignado en la iteración previa y si la nueva distancia es menor o igual que la distancia previa entonces el objeto se descarta de cálculos subsecuentes.

Otros trabajos que siguen este mismo enfoque son [16] y [17], aunque en [18] se propone un umbral para cada centroide obtenido del cálculo de distancia entre centroides. En [19] abordan también la idea de identificar los objetos que no cambiarán de grupo en iteraciones subsecuentes, sin embargo, los autores proponen además comprimir y eliminar dichos objetos para reducir la cantidad de cálculos en las siguientes iteraciones.

II. La metaheurística N-means

N-means se compone de dos heurísticas denominadas H1 y H2. La descripción de las heurísticas se explica a continuación.

A. Heurística H1

La heurística H1 surge a partir de las observaciones en el comportamiento del algoritmo K-means cuando resuelve instancias sintéticas con distribucion uniforme, de entre las cuales se destacan las siguientes:

- Los objetos que están cercanos a los centroides tienen baja posibilidad de cambiar de grupo en iteraciones subsecuentes.
- Los objetos que son equidistantes a sus dos centroides más cercanos pueden asignarse a alguno de los dos grupos representados por dichos centroides.
- Los objetos cuasi-equidistantes de sus dos centroides más cercanos tienen una alta posibilidad de cambiar de grupo en iteraciones subsecuentes.
- Un factor determinante por el cual los objetos cambian de grupo es el desplazamiento de los centroides en cada iteración.
- Durante el desplazamiento de los centroides a través de diferentes iteraciones, aproximadamente la mitad de los objetos se encontrarán a una distancia más corta respecto a la nueva posición del centroide, mientras que la otra mitad de los objetos estarán a una distancia más larga. Entre más alejados se encuentren los objetos de la nueva posición del centroide, más alta es la posibilidad de que el objeto cambie de grupo en las iteraciones posteriores.

Por lo tanto, la heurística H1 define dos subconjuntos para cada uno de los grupos. El primer subconjunto se compone de aquellos objetos que presentan baja probabilidad de cambio de grupo, debido a que tales objetos se encuentran cercanos a su centroide. El segundo subconjunto se compone de aquellos objetos que tienen alta posibilidad de cambio de grupo y por lo tanto, siguen participando en el cálculo de distancias objeto-centroides.

La heurística H1 requiere dos definiciones. La primera es el índice de equidistancia (α_i) que se define como la diferencia de las distancias entre cada objeto y sus dos centroides más cercanos, en la iteracion i. La segunda es el umbral de equidistancia (β_j) que se define como la suma de los dos desplazamientos mayores de centroides entre la iteración $(j-1)$ y la iteración j. Si se satisface la condición $(\alpha_i > \beta_j)$ entonces el objeto i se asigna de forma temprana en la iteración j. Para conocer mayores detalles de la heurística H1 se invita al lector a consultar [20].

B. Heurística H2

Derivado de las ejecuciones del algoritmo K-means se observa en su comportamiento que algunos grupos son estables en iteraciones tempranas, esto es, los grupos estables ya no tienen intercambio de objetos con los grupos vecinos. Se entiende como iteraciones tempranas a aquellas que se realizan antes de la mitad del número total de iteraciones. En consecuencia se introduce esta heuristíca en el paso de clasificación.

La nueva heurística, denominada H2, establece que los objetos asignados a un grupo estable pueden ser descartados de los cálculos de distancia en las iteraciones posteriores.

En la Figura 1 se muestran seis iteraciones del algoritmo K-means cuando resuelve una instancia de 10,000 objetos divididos en 100 grupos y con una distribución uniforme de dos dimensiones en un espacio de 100 x 100.

La Figura 1(a) muestra cómo el grupo identificado con el número 1 se vuelve estable en la iteración 8, es decir, deja de intercambiar objetos entre los grupos vecinos. Cuando un grupo deja de intercambiar objetos su centroide no cambia hasta alcanzar el criterio de convergencia del algoritmo.

En la Figura 1(b) se observa que en la iteración 9 dos grupos nuevos obtienen la condición estable (grupos 2 y 3); la Figura 1(c) muestra que en la iteración 10 el número de grupos que alcanzan la condición estable son 8 (grupos 4-11).

Las siguientes tres iteraciones de la Figura 1 (d, e y f) representan las iteraciones finales del algoritmo. En el caso de la Figura 1(d) se observa que sólo 12 grupos no han alcanzado la condición estable, esto es, aquellos grupos que carecen de un número identificador. La figura 1(e) muestra que sólo 3 grupos no han obtenido la condición estable y finalmente la Figura 1(f) muestra a todos los grupos estables y, por consiguiente, el agrupamiento termina (todos los grupos cuentan con un número identificador).

Derivado de las observaciones del comportamiento de las iteraciones de la Figura 1, se establece cómo se pueden excluir los objetos (de los grupos estables) del cálculo de distancias en iteraciones subsecuentes. Por lo tanto, es posible reducir la complejidad del algoritmo y se espera que el comportamiento del algoritmo sea asintótico, es decir, en cada iteración converge hacia la solución.

III. Experimentación y discusión

Para validar las heurísticas propuestas se siguió un enfoque empírico y la metodología experimental está inspirada en [21]. Se realizó la implementación del algoritmo y se midió su desempeño con dos elementos: tiempo de ejecución y calidad de la solución.

El tiempo de ejecución se mide como el tiempo transcurrido, desde la primera iteración, entre la asignación de cada objeto a su centroide más cercano y hasta la última iteración que satisfaga el criterio de convergencia del algoritmo. El criterio de paro es aquel en el que los centroides no cambian entre la iteración actual y la previa. Cabe señalar que estas condiciones fueron las mismas durante la ejecución de K-means estándar y N-means.

A. Proceso experimental

1) Formulación de la pregunta:
   ¿Que tanto reduce N-means la complejidad de K-means y que tanto afecta su calidad? El algoritmo K-means presenta una complejidad $O(nkd)$ por cada iteración, lo que significa que se tienen 3 parámetros que impactan en su

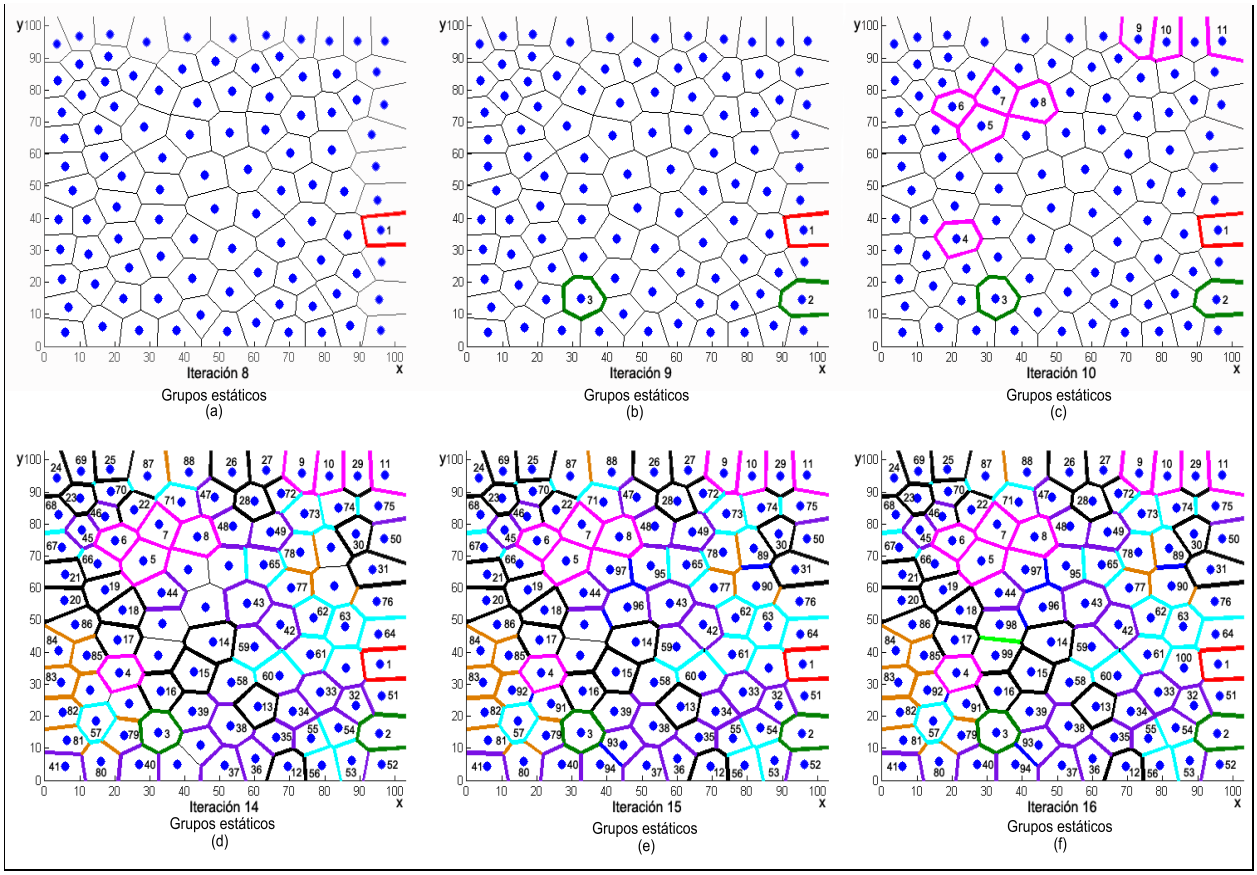

Fig. 1. Comportamiento visual del algoritmo K-means cuando resuelve una instancia de 10,000 objetos divididos en 100 grupos

complejidad (número de objetos (n), números de grupos (k) y número de dimensiones (d), respectivamente).

2) Entorno de prueba:
K-means estándar y N-means se programaron en lenguaje C. El equipo utilizado tiene la siguiente configuración: Procesador Intel Core 2 Duo T6400 2.0GHz, 4GB en RAM y 500GB en HD. El sistema operativo es Linux (Ubuntu) 13.04.
3) Instancias de prueba:
Los datasets provienen del repositorio de la Universidad Eastern Finland [22] y son los siguientes: Birch1 con 100,000 objetos en 2 dimensiones (sintético) y el conjunto de datasets DIM que se compone de 6 instancias de 1,024 objetos cada uno y con 32, 64, 128, 256, 512 y 1,024 dimensiones respectivamente (sintéticos).
4) Diseño del experimento:
Para la realización de los experimentos se consideran los parámetros (n, k, d) realizando incrementos en cada uno de ellos. Por ejemplo, al variar n, ¿Cómo cambia la relación de disminución de la complejidad entre K-means y N-means?
Por lo tanto se realizaron 3 tipos de experimentos para observar cómo afecta el cambio de objetos, de grupos y de dimensiones.
El experimento A tiene como propósito ver cómo se comporta K-means y N-means cuando se incrementa el número de objetos. Se tiene como base el dataset Birch1 y se obtuvieron 9 muestras. La muestra 1 de 10,000 objetos, la muestra 2 de 20,000 objetos y así sucesivamente hasta alcanzar los 90,000 objetos. Por último, el propio dataset Birch1 representa la muestra con el número total de objetos. El número de grupos se mantuvo fijo para todos los casos $(k = 16)$ al igual que el número de dimensiones $(d = 2)$.
El experimento B tiene como objetivo ver cómo se comporta K-means y N-means a medida que cambia el número de grupos. Este experimento también utiliza el dataset Birch1 y, para este caso, se resolvieron 7 instancias del dataset con los siguientes valores para el número de grupos: 2, 4, 8, 16, 32, 64 y 128. El número de objetos se mantuvo fijo para todos los casos $(n = 100,000)$ así como el número de dimensiones $(d = 2)$.
Por último el experimento C tiene como propósito ver cómo se comporta K-means y N-means cuando se incrementa el número de dimensiones. Para este caso se utiliza el dataset DIM y se resolvieron 6 instancias con los siguientes valores: 32, 64, 128, 256, 512 y 1024. El número de objetos se mantuvo fijo para todos los casos $(n = 1024)$ así como también el número de grupos

$(k = 16)$.

5) Ejecución de los experimentos:
   El número de ejecuciones de cada experimento es de 30. La generación de los centroides iniciales es aleatoria y diferente para cada ejecución. Sin embargo, los mismos centroides iniciales son utilizados por K-means y N-means.

B. Resultados experimentales

Las Figuras 2, 3 y 4 muestran respectivamente como a medida que aumentan el número de objetos, el número de grupos y el número de dimensiones, se incrementa de manera no lineal el tiempo de ejecución de K-means. En relación con N-means es destacable que tiende a tener un comporamiento lineal.

Las Tablas 1, 2 y 3 muestran los resultados promedio de las 5 ejecuciones de cada experimento. En la Tabla 1 se contrastan los resultados de calidad entre K-means y N-means cuando se resuelven instancias con diferente número de objetos (experimento A). En la primera columna se muestra el número de objetos de la instancia, en la segunda y tercera columna se expresan los valores de error al cuadrado y la cuarta columna indica la diferencia de calidad en porcentaje. Se observa que en el peor de los casos N-means disminuye la calidad en un 5.5% con relación a la calidad de K-means.

En la Tabla 2 se contrastan los resultados de calidad entre K-means y N-means al resolver instancias con diferente número de grupos (experimento B). En la primera columna se muestra el número de grupos de cada instancia, en la segunda y tercera columna se expresan los valores de error al cuadrado y la cuarta columna indica la diferencia de calidad en porcentaje. En el peor de los casos N-means disminuye la calidad tan sólo en un 5.3% con relación a la calidad de K-means.

En la Tabla 3 se contrastan los resultados de calidad entre K-means y N-means cuando se resuelven instancias con diferente número de dimensiones (experimento C). En la primera columna se muestra el número de dimensiones de cada instancia, en la segunda y tercera columna se expresan los valores de error al cuadrado y la cuarta columna indica la diferencia de calidad en porcentaje. Se observa que en todos los casos N-means disminuye la calidad en no más de 0.002% con relación a la calidad de K-means.

Objetos	Error K-means	Error N-means	Diferencia (%)
10,000	877,768,462.54	899,666,865	-2.49
20,000	1764,792,539.11	1846,506,317	-4.63
30,000	2639,132,438.07	2724,070,892	-3.21
40,000	3552,053,504.18	3727,674,126	-4.94
50,000	4406,016,072.59	4569,932,212	-3.72
60,000	5316,492,341.70	5469,576,797	-2.87
70,000	6191,305,736.44	6363,612,376	-2.78
80,000	7075,926,111.29	7329,171,466	-3.57
90,000	7936,888,487.95	8377,966,020	-5.55
100,000	8789,977,031.60	9094,510,610	-3.46

TABLE I
RESULTADOS DE CALIDAD DEL EXPERIMENTO A

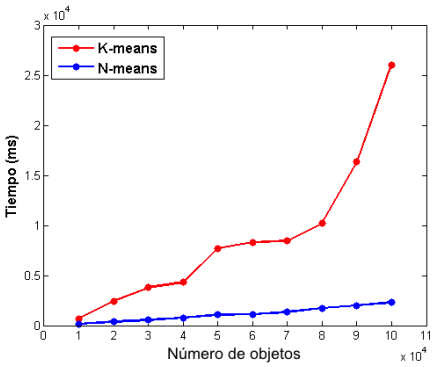

Fig. 2. Resultados y tendencias al incrementar el número de objetos

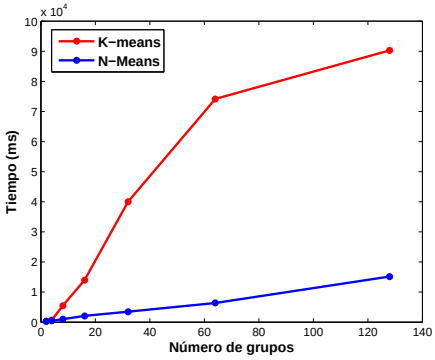

Fig. 3. Resultados y tendencias al incrementar el número de grupos

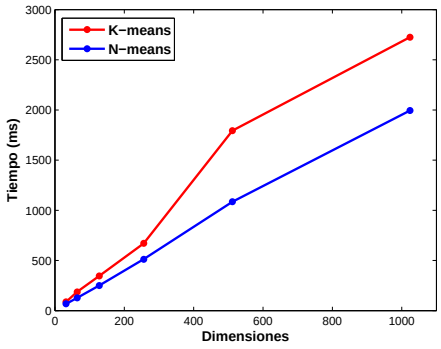

Fig. 4. Resultados y tendencias al incrementar el número de dimensiones

Grupos	Error K-means	Error N-means	Diferencia (%)
2	27,285,157,378.49	27,318,858,061.42	-0.12
4	17,537,487,160.47	17,832,352,689.05	-1.68
8	12,613,529,746.02	12,750,752,381.12	-1.08
16	8,858,756,141.81	9,335,704,671.28	-5.38
32	6,234,648,561.73	6,311,130,342.45	-1.22
64	4,245,103,610.57	4,374,625,397.65	-3.05
128	2,613,726,055.42	2,714,424,969.49	-3.85

TABLE II
RESULTADOS DE CALIDAD DEL EXPERIMENTO B

IV. CONCLUSIONES

En este trabajo se muestra que es factible reducir la complejidad del algoritmo K-Means de manera importante,

Dimensiones	Error K-means	Error N-means	Diferencia (%)
32	100,243.85	100,244.62	-0.00076
64	170,238.29	170,238.31	-0.00001
128	149,376.86	149,380.19	-0.00223
256	158,619.62	158,623.23	-0.00227
512	527,591.83	527,592.35	-0.00009
1024	696,589.16	696,590.84	-0.00024

TABLE III
RESULTADOS DE CALIDAD DEL EXPERIMENTO C

mediante el uso de la meta-heurística propuesta, a la cual se le denominó N-means. Con base en los experimentos realizados se observó que con N-Means se obtuvieron reducciones de tiempo hasta de un 91% y una disminución de la calidad de sólo 5.5%. Es destacable que con base en el análisis de los resultados se observó un comportamiento cuasi lineal de N-means. Esta particularidad cobra importancia cuando se resuelven instancias con un gran número de objetos, de grupos o de dimensiones de los datos.

REFERENCES

[1] R. Xu and D. C. Wunsch, *Clustering*. IEEE Press - John Wiley & Sons, 2008.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, 1967, pp. 281–296.

[3] S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[4] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine learning*, vol. 56, pp. 9–33, 2004.

[5] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[6] J. Wu, *Advances in K-means Clustering: A Data Mining Thinking*. Springer, 2012.

[7] M. E. Agha and W. M. Ashour, "Efficient and fast initialization algorithm for k-means clustering," *International Journal of Intelligent Systems and Applications*, vol. 1, no. 1, pp. 21–31, 2012.

[8] A. H. Ahmed and W. Ashour, "An initialization method for the k-means algorithm, using rnn and coupling degree," *International Journal of Computer Applications*, vol. 25, no. 1, pp. 1–6, 2011.

[9] M. B. Al-Daoud, "A new algorithm for cluster initialization," in *WEC'05: The Second World Enformatika Conference*, Istanbul, Turkey, 2005, pp. 74–76.

[10] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for k-means clustering," *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1293–1302, 2004.

[11] M. F. Eltibi and W. M. Ashour, "Initializing k-means clustering algorithm using statistical information," *International Journal of Computer Applications*, vol. 29, no. 7, pp. 51–55, 2011.

[12] S. J. Redmond and C. Heneghan, "A method for initializing the k-means clustering algorithm using kd-trees," *Pattern Recognition Letters*, vol. 28, no. 8, pp. 965–973, 2007.

[13] C. S. Li, "Cluster center initialization method for k-means algorithm over data sets with two clusters," *Procedia Engineering*, pp. 324–328, 2011.

[14] X. Zhanguo, C. Shiyu, and Z. Wentao, "An improved semi-supervised clustering algorithm based on initial center points," *Journal of Convergence Information Technology*, vol. 7, no. 5, pp. 317–324, 2012.

[15] A. Fahim, A. Salem, F. Torkey, and M. Ramadan, "An efficient enhanced k-means clustering algorithm," *Journal of Zhejiang University SCIENCE A*, vol. 7, no. 10, pp. 1626–1633, 2006.

[16] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on*, 2010, pp. 63–67.

[17] O. Sangita and J. Dhanamma, "An improved k-means clustering approach for teaching evaluation," in *Advances in Computing, Communication and Control*. Springer, 2011, pp. 108–115.

[18] R. V. Singh and M. Bhatia, "Data clustering with modified k-means algorithm," in *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, 2011, pp. 717–721.

[19] C.-W. Tsai, C.-S. Yang, and M.-C. Chiang, "A time efficient pattern reduction algorithm for k-means based clustering," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 504–509.

[20] J. Pérez, C. E. Pires, L. Balby, A. Mexicano, and M. Hidalgo, "Early classification: A new heuristic to improve the classification step of k-means," *Journal of Information and Data Management*, vol. 4, no. 2, pp. 94–103, 2013.

[21] C. C. McGeoch, *A guide to experimental algorithmics*. Cambridge University Press, 2012.

[22] "Clustering datasets," http://cs.joensuu.fi/sipu/datasets/, July 2014.