

Instituto Tecnológico Autónomo de México



Proyecto Final Métodos Analíticos

---

**Mercados Educativos en México**

---

*Profesor:*  
Felipe Gonzalez

Sebastián Cadavid Sánchez 191070 - Paola Mejía Domenzaín 157093

26 de mayo de 2020

# Índice

<b>1. Descripción del problema a resolver</b>	<b>2</b>
<b>2. Objetivos a alcanzar</b>	<b>2</b>
<b>3. ¿Qué se está haciendo actualmente?</b>	<b>2</b>
<b>4. Planteamiento del problema</b>	<b>4</b>
<b>5. Métricas a usar</b>	<b>4</b>
<b>6. Descripción del Baseline</b>	<b>4</b>
<b>7. Solución que se propone para el proyecto</b>	<b>5</b>
7.1. Construcción de tabla de <i>switchers</i> . . . . .	5
7.2. Creación de <i>commuting areas</i> . . . . .	6
7.3. <i>Commuting areas</i> utilizando <i>buffers</i> . . . . .	6
7.4. Detección de comunidades . . . . .	7
<b>8. Resultados</b>	<b>8</b>
8.1. <i>Commuting areas</i> . . . . .	8
8.2. Mercados . . . . .	9
<b>9. Conclusiones y recomendaciones</b>	<b>13</b>
<b>A. Código Switchers Primaria y secundaria</b>	<b>15</b>
<b>B. Construcción de grafo</b>	<b>17</b>
<b>C. Construcción de commuting zones</b>	<b>18</b>
<b>D. Comparación de algoritmos</b>	<b>19</b>
<b>E. Cálculo de modularidad</b>	<b>20</b>

## **1. Descripción del problema a resolver**

El problema a resolver es encontrar los mercados educativos a nivel secundaria y primaria en México. Un mercado, en términos económicos, está compuesto por la oferta y la demanda por un bien o servicio, y estas dos facciones determinan el precio por este último. En un mercado educativo, la oferta está compuesta por escuelas que ofertan educación y por estudiantes que demandan dicho servicio. Sin embargo, las características específicas de los agentes (geográficas y socio-económicas), implican que haya segmentación en los mercados de este tipo.

Delimitar tales mercados educativos permite estudiar las relaciones entre los centros educativos y comprender las dinámicas que rigen al sector educativo en niveles de baja agregación. Asimismo, proporciona información sobre las decisiones y preferencias de los padres y estudiantes en la elección por escuelas.

Sin embargo, construir una definición para acotar este tipo de mercados no es trivial. Este proyecto presenta una de muchas posibles alternativas enfocándose en las decisiones de los estudiantes (la demanda) y analiza los resultados.

## **2. Objetivos a alcanzar**

El objetivo de este proyecto es construir mercados educativos de educación primaria y secundaria en México. Para lograr lo anterior, los objetivos secundarios son crear una definición de mercado y documentar las suposiciones; además de explorar y comparar diferentes métodos de detección de comunidades.

El reporte se concentrará en los resultados de los mercados de secundaria pero se utilizará la misma metodología para primaria.

## **3. ¿Qué se está haciendo actualmente?**

En el ámbito de políticas públicas, los mercados se han utilizado para estudiar la segregación y desigualdades sociales [1] y los incentivos para que las escuelas inviertan en calidad [2]. Por otro lado, la definición del concepto de mercados educativos es relativamente reciente en la literatura. En particular, para nuestro conocimiento, existen dos trabajos previos a este que buscan responder preguntas similares.

Inicialmente, [2] estudia para Chile los efectos de la estructura de diseño de políticas de cupones sobre incentivos de escuelas, y también los efectos asociados en la distribución del rendimiento académico entre distintos grupos socio-económicos. Para realizar este análisis, este autor construye mercados educativos teniendo en cuenta la distancia como una de las características centrales.

En particular, este autor resalta la importancia de definir fronteras de mercado más allá de las territoriales administrativas. Por ejemplo, en algunas zonas donde hay Estados con alta interco-

nexión (i.e proximidad espacial), es factible que algunos estudiantes vivan en el Estado A, y por ejemplo, estudien en el Estado B. (La razones por las cuales se generen estos comportamientos pueden ser varias, y no únicamente la cercanía geográfica. Por ejemplo, la diferencia en la distribución de precios entre Estados.) En este sentido, si se realiza una partición administrativa, se pierde la captación de este tipo de comportamientos.

La propuesta realizada por [2] para abordar este problema en la construcción geográfica de los mercados incluye los siguientes procedimientos: i) se toman como base las divisiones territoriales administrativas; ii) se unen las zonas urbanas que no tienen una distancia superior a 2 kms; iii) se construyen *buffers* de 1 km alrededor de la frontera de las zonas definidas en ii). Con esta metodología, el autor obtiene un total de 363 mercados que incluyen aproximadamente 4500 escuelas. Donde el mercado más grande incluye más de 1000 escuelas, y los más pequeños, una sola escuela.

Por otro lado, [3] realizan un estudio (en progreso) donde evalúan el efecto de las políticas educativas en Colombia con relación a la retención de estudiantes por parte de las escuelas dados diferentes factores de criminalidad. Para lograr estudiar estos efectos, estos autores también realizan un análisis granular a nivel de mercados. La definición de mercados educativos utilizó modelos de grafos y construyó en dos pasos: i) se computan *commuting areas*<sup>1</sup> utilizando los datos de municipio de la residencia de los estudiantes y de las escuelas; ii), dentro de esas *commuting areas*, se usaron métodos de agrupación de aristas en los cada grafo construido con las escuelas dentro de la *commuting area* [3].

En detalle, para calcular los pesos en las artistas, se construyeron matrices de adyacencia que dicen cuántos alumnos tienen en común cada par de colegios (para un periodo dado de tiempo y restringiendo la muestra a alumnos que se cambian entre escuelas (*switchers*)).

Utilizando la matriz de adyacencia, se obtuvieron redes con las cuales se extrajeron los mercados utilizando la función *getLinkCommunities* del paquete *linkcomm* de R. La función utiliza el coeficiente de Tanimoto (ver ecuación 1) para obtener la similitud entre los aristas [4].

$$S(e_{ik}, e_{jk}) = \frac{\mathbf{a}_i \cdot \mathbf{a}_j}{|\mathbf{a}_i|^2 + |\mathbf{a}_j|^2 - \mathbf{a}_i \cdot \mathbf{a}_j} \quad (1)$$

donde  $\mathbf{a}_i$  corresponde a un vector que describe los pesos de los enlaces entre el nodo  $i$  y los nodos en las vecindades de primer orden de los nodos  $i$  y  $j$ . Esta medida es igual a 0 en caso de que no haya dicha conexión).

Una dificultad para implementar este método con los datos de México es que no están disponibles los domicilios de los alumnos, por lo tanto, resulta difícil construir las *commuting areas*. Asimismo, una de las limitaciones de utilizar el método que utilizaron [3] para Colombia para agrupar los aristas es que el algoritmo falla en *commuting areas* con menos de 20 escuelas.

Nuestro enfoque para abordar la creación de *commuting areas* en México plantea dos posibilidades. En la primera, utilizamos la metodología utilizada por [2] para la creación de estas zonas territoriales. En la segunda, seguimos el enfoque de [3] utilizando grafos. La creación de

---

<sup>1</sup>Territorios en los cuales gente con ciertas características socioeconómicas y geográficas en común se desplaza diariamente.

mercados educativos siempre se realiza con métodos de grafos.

## 4. Planteamiento del problema

En términos de lo visto en clase, el problema se traduce en un análisis de detección de comunidades. Inicialmente, el problema es crear una base de datos con información de los estudiantes y las escuelas en México. Posteriormente, se pueden acotar las zonas donde pueden existir las comunidades. Finalmente, se puede construir un grafo y utilizar algoritmos de agrupamiento para detectar las comunidades (mercados educativos).

## 5. Métricas a usar

La métrica a utilizar para evaluar los diferentes algoritmos de agrupamiento será la modularidad. La modularidad (*modularity*) se diseña para medir la fuerza de una división de una red en comunidades. Es decir, intentar cuantificar qué tan buena o “cohesiva” es una separación de nodos en grupos [5].

La modularidad de una gráfica no dirigida y vértices con una agrupación dada  $g$  se define como

$$Q = \frac{1}{2m} \sum_{u,v} \left( A_{u,v} - \frac{k(u)k(v)}{2m} \right) I(g(u), g(v)) \quad (2)$$

donde  $A$  es la matriz de adyacencia y  $k(u)$  es el grado de  $u$ .

Es importante remarcar que la modularidad tiene un límite de resolución y como consecuencia, no es capaz de detectar comunidades pequeñas.

En particular, elegimos esta métrica de evaluación, en la medida que da un indicio de la solidez de la conexión entre escuelas (nodos) al interior de los mercados educativos (comunidades). En este sentido, nos permite ver qué tan buenos son los mercados que se están generando. Lo anterior, en la medida que, medidas de mayor modularidad implican que las escuelas que están en el mismo mercado están altamente interconectadas, y por lo tanto, los alumnos si rotan de manera razonable entre ellas.

## 6. Descripción del Baseline

El baseline es definir los mercados basándose únicamente en el municipio al que pertenece la escuela. Es decir, asignar arbitrariamente todas las escuelas de un municipio a un mercado.

Con esta definición, se tienen 2,457 mercados en el cual el mercado más grande tiene 292 escuelas y el más chico tiene una escuela. La modularidad de esta agrupación es 0.76.

## 7. Solución que se propone para el proyecto

Lo ideal para construir los mercados educativos sería “entrar” en la cabeza de los estudiantes y de los padres para conocer sus preferencias de escuelas, y por lo tanto, entender las dinámicas del cambio entre escuelas de los alumnos.

Como alternativa, podemos “inferir” las preferencias con los alumnos que se cambiaron de escuela (*switchers*), a partir del monitoreo geográfico de sus cambios de institución.

### 7.1. Construcción de tabla de *switchers*

El objetivo de esta tabla es saber cuántos alumnos se cambiaron de la escuela A a la escuela B en un periodo de tiempo.

Para construirla, se utilizaron las bases de datos de la Evaluación Nacional de Logros Académicos en Centros Escolares (ENLACE). En esta base, cada alumno se identifica únicamente por su Clave Única de Registro de Población (CURP) y sabemos la Clave de Centro de Trabajo (CCT). Utilizando los datos desde el 2006 hasta el 2013 (7 años) se identificaron los alumnos que cambiaron de escuela.

Una de las dificultades de los datos es que si un estudiante faltó el día que se presentó la prueba, para los datos considerados, no hay manera de saber en qué escuela estaba en ese año. Sin embargo, en vez de buscar solo en el año consecutivo, buscamos en todos los años posteriores. Es decir, si la estudiante Paola estuvo en la escuela A en el año 2006 y no presentó la prueba en el año 2007, entonces la buscamos en el año 2008 o en años consecutivos hasta encontrarla. Si encontramos a Paola en la escuela B en el año 2010 entonces registramos que se cambió entre la escuela A y la escuela B. Es importante notar que, es posible que Paola se hubiera cambiado a la escuela C en el 2008 antes de haber entrado a la escuela B; sin embargo, esta información no se capturó.

Cabe resaltar que para primaria solo 2 millones de alumnos de los 40 millones presentaron un cambio de escuela, y por tanto los capturamos como *switchers*. Es decir, los mercados se construyeron con las preferencias el 5 % de los estudiantes.

La tabla 1 muestra un ejemplo del formato de la tabla de *switchers*.

Escuela Origen	Escuela Destino	Número de switchers
A	B	32
B	A	2
C	B	12

**Tabla 1:** Ejemplo de tabla de *switchers*

## 7.2. Creación de *commuting areas*

Es posible que si se toman en cuenta únicamente criterios de migración de estudiantes de una escuela hacia otra, se generen conexiones entre escuelas que no necesariamente impliquen que estas pertenezcan al mismo mercado educativo. Por ejemplo, consideremos el caso de un estudiante que se muda de Ciudad de México a Monterrey, y por tanto, se cambia de escuela.

En general, esperaríamos que los mercados educativos se encuentren en zonas geográficas cercanas, donde parezca razonable que se abarquen localidades donde gente con ciertas características en común se desplace diariamente. Lo anterior, corresponde al concepto de *commuting areas*. En este trabajo se delimitan las zonas donde pueden estar los mercados educativos con respecto al anterior concepto.

La creación de subgrafos puede realizarse de distintas maneras. En este trabajo abordamos dos posibles enfoques: 1) computando *buffers*<sup>2</sup> alrededor de escuelas para las *commuting areas* y 2) utilizando grafos para los mercados.

## 7.3. *Commuting areas* utilizando *buffers*

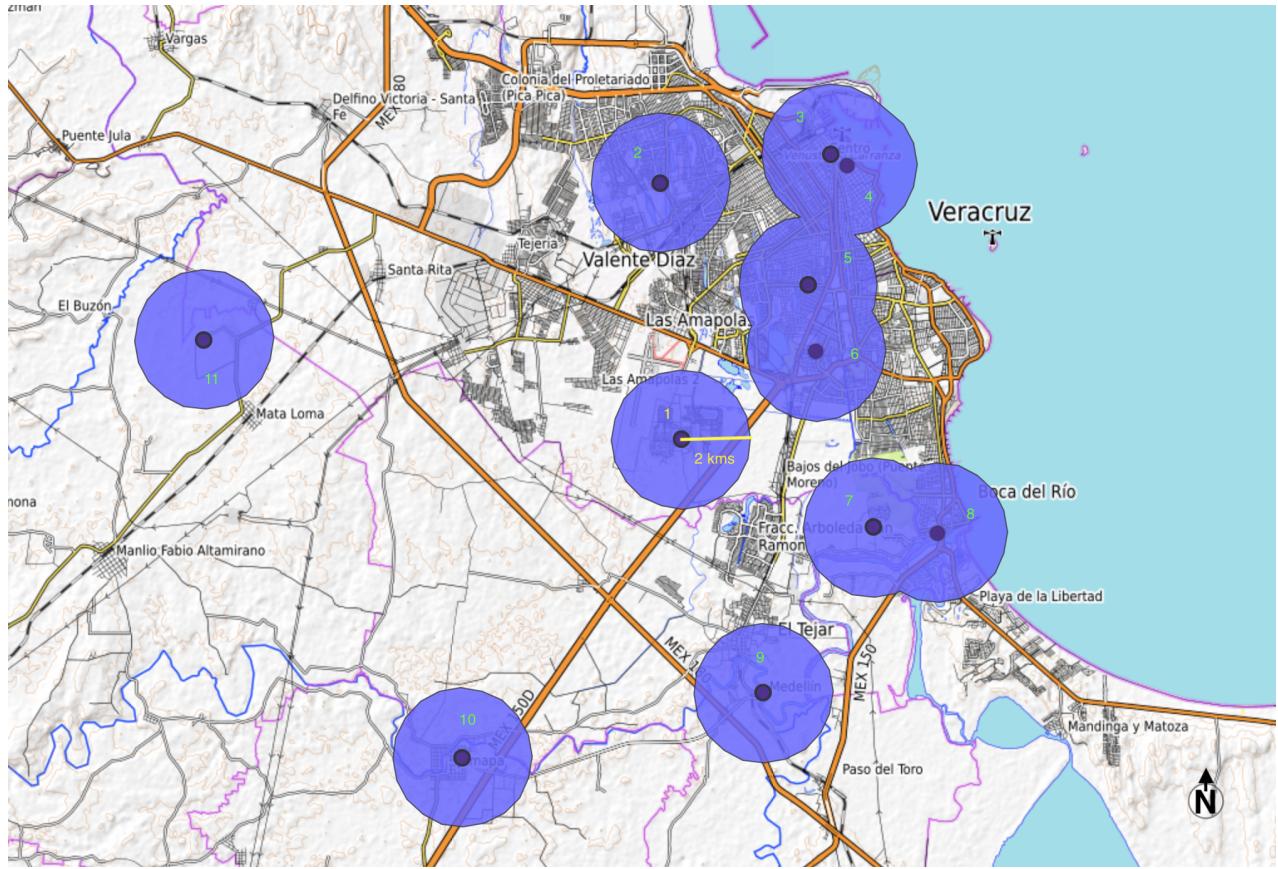
Un *buffer*, es una área que se define a partir de una estructura espacial. En este caso, son de tipo circular con un radio definido en kilómetros y se forman a partir de las coordenadas de cada escuela. En particular, para esta implementación utilizamos la proyección UTM zona 15N con EPSG 6370<sup>3</sup> y se evalúa el computo de *buffers* desde 1 km hasta 15 kms (con variaciones de 0.5 kms).

A modo de ejemplo, considere la figura 1, en la cual se dibujan los *buffers* asociados a 11 escuelas de secundaria en Veracruz.

---

<sup>2</sup>Regiones alrededor de objetos espaciales, cuyos puntos generalmente equidistan de los elementos individuales que lo constituyen. En este ejercicio corresponde a una región circular con un radio en kilómetros definido.

<sup>3</sup>Ver: <https://epsg.io/6370>



**Figura 1:** Mapa de Veracruz con la ubicación de 11 escuelas de secundaria. Las áreas demarcadas en azul translúcido denotan *buffers* circulares de 2 kms. Los puntos oscuros denotan la ubicación de la respectiva escuela.

Una vez se han computado los respectivos *buffers* para la totalidad de escuelas en el conjunto de datos, se procede a identificar cuales están sobrepuertos y comparten intersecciones, y por lo tanto, como conjunto, conforman una *commuting areas*. Como se puede notar en la figura 1, esto sucede para los conjuntos  $\{3, 4, 5, 6\}$  y  $\{7, 8\}$ . Lo que implica que en total se tienen 7 *commuting areas*. El anterior proceso es extrapolado para todo México (código en apéndice C).

## 7.4. Detección de comunidades

La idea principal fue construir un grafo no dirigido en el cual los nodos son las escuelas y los pesos en los aristas son el flujo de alumnos entre las escuelas.

El primer paso fue construir el grafo. Para encontrar el peso de los aristas se sumaron los alumnos que se cambiaron en ambas direcciones como se ve en el apéndice ?? y se dividió el flujo entre el total de alumnos que se cambiaron entre escuelas como se muestra en la ecuación

(3):

$$weight = \frac{\text{cambio de A a B} + \text{cambio de B a A}}{\text{cambios fuera de A} + \text{cambios fuera de B}} \quad (3)$$

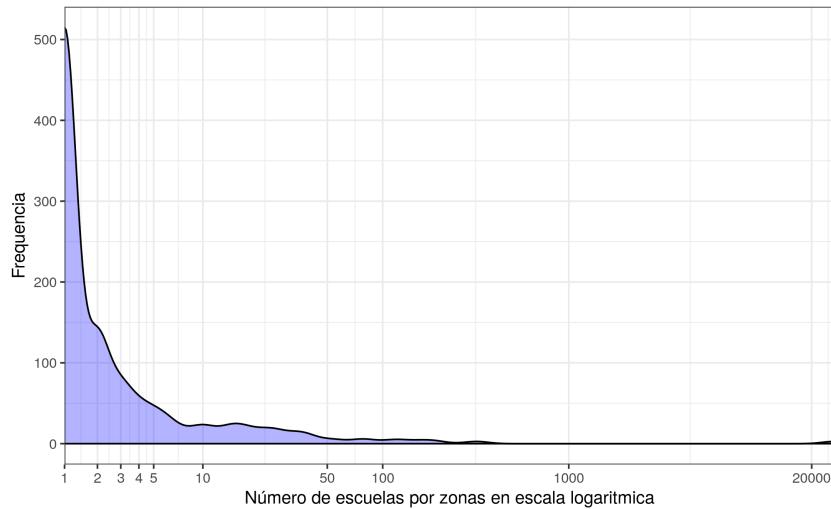
A continuación, se compararon los siguientes algoritmos de detección de comunidades (código en apéndice D):

- *leading eigenvector*: El algoritmo funciona calculando los vectores propios de la matriz de modularidad para el valor propio positivo más grande y después separando los vértices en dos comunidades basado en el signo del vector propio [6]. En el paquete de R igraph está bajo el nombre cluster\_leading\_eigen.
- *label propagation*: El algoritmo funciona etiquetando los vértices con etiquetas únicas y actualizando las etiquetas con los votos de los vértices vecinos. Es decir, si una escuela está rodeada por cinco escuelas de la comunidad A y tres escuelas de la comunidad B, entonces por mayoría de votos la escuela va a pertenecer a la comunidad A [7]. En el paquete de R igraph está bajo el nombre cluster\_label\_prop.
- *walktrap*: El algoritmo encuentra subgrafos densamente conectados por medio de caminatas aleatorias bajo el supuesto que caminatas aleatorias pequeñas tienen a permanecer en la misma comunidad [8]. En el paquete de R igraph está bajo el nombre cluster\_walktrap.
- *fast greedy*: Toma un enfoque jerárquico en el cual inicialmente cada nodo es asignado a una comunidad individual y en cada paso los vértices son re-asignados a la comunidad que genere la mayor contribución a la modularidad [9]. En el paquete de R igraph está bajo el nombre cluster\_fast\_greedy.
- *multi level*: Es muy similar al algoritmo miope (fast greedy) pero incluye un paso de agregación de comunidades que resulta útil para redes muy grandes. Es decir, después de re-asignar los vértices, como en el algoritmo fast greedy, cada comunidad se considera como un vértice y el proceso vuelve a comenzar [10]. En el paquete de R igraph está bajo el nombre cluster\_louvain.

## 8. Resultados

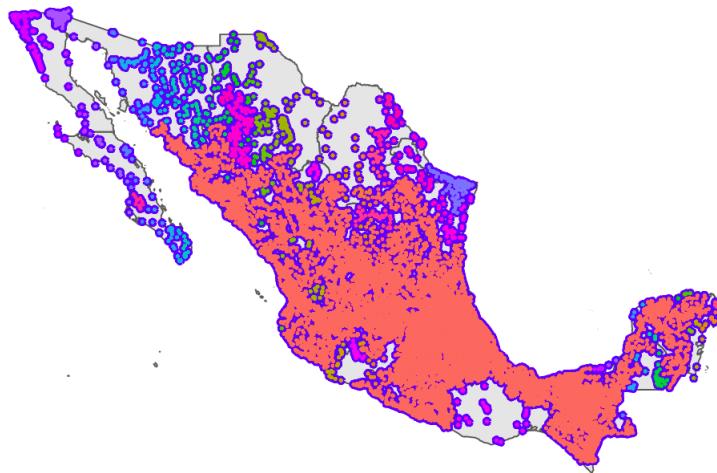
### 8.1. Commuting areas

Se escogió utilizar buffers de 10 km. Como resultado, obtuvimos 326 diferentes grupos. Dentro de estos grupos, el número promedio de escuelas por grupo es 88 aunque como se ve en la figura 2 la distribución no es uniforma ya que el grupo con más escuelas tiene 26,406 escuelas y el grupo con menor número de escuelas tiene 1.



**Figura 2:** Distribución de número de escuelas por commuting zone

La figura 3 muestra las diferentes commuting zones en México.



**Figura 3:** Visualización de commuting zones en México

## 8.2. Mercados

Dentro de cada *commuting zone* con más de una escuela, se utilizaron los cinco algoritmos mencionados previamente para detectar comunidades. A continuación, se comparó la modularidad de los diferentes algoritmos para cada bloque y se eligió un ganador. La tabla 2 muestra el número de veces que cada algoritmo tuvo la mayor modularidad dentro de una *commuting zone*.

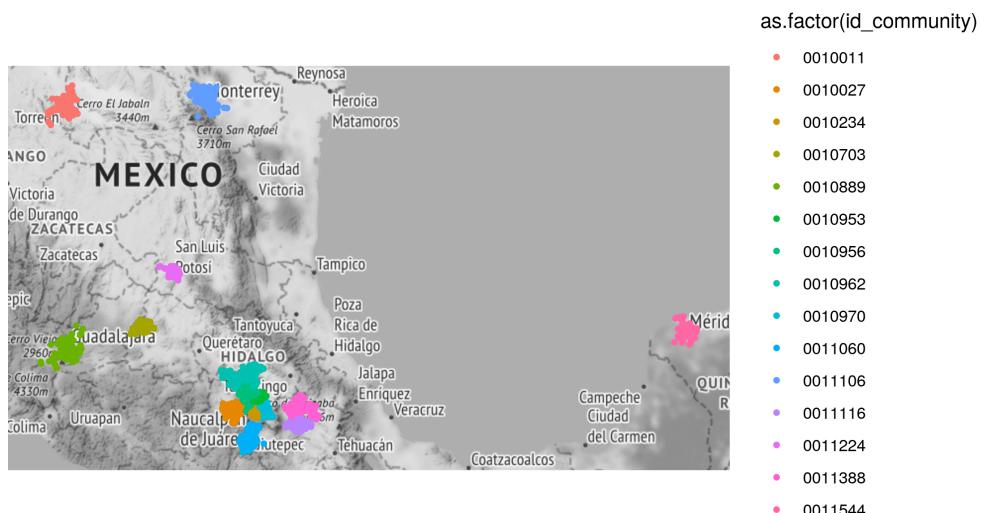
algoritmo	wins
fast greedy	27
multi level	14
walktrap	14
label propagation	6
leading eigenvector	3

**Tabla 2:** Resumen de algoritmos con mayor modularidad

A continuación, se crearon los mercados dentro de cada *commuting zone* utilizando el algoritmo que maximizara la modularidad en cada zona. Finalmente, se creó una columna de *community id* en la cual los primeros tres dígitos se usan para identificar a la *commuting zone* y los últimos cuatro dígitos identifican el mercado dentro de cada una. Por ejemplo, la comunidad 0010027 es la comunidad 27 del *commuting zone* 1. Utilizando el *community id* como identificador de la comunidad la modularidad del grafo fue de 0.91 (código en apéndice E, lo cuál superó el baseline de utilizar los municipios de 0.76).

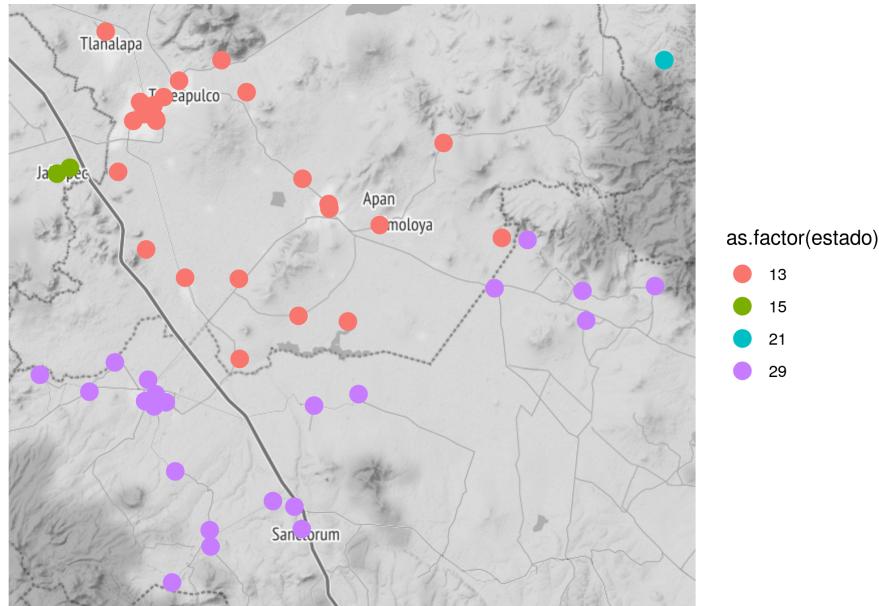
La figura 4 muestra las 15 comunidades más grandes. Es interesante comparar esta figura con la figura 3 en la cual había un grupo que parecía abarcar todas las escuelas. La *commuting zone* más grande tenía 26,406 escuelas mientras que el mercado o comunidad más grande costa de 813 escuelas.

Asimismo, los mercados educativos en su mayoría corresponden a las ciudades más grandes como Nuevo León, Guadalajara y Mérida. Sin embargo, es relevante notar que uno de los mercados más grandes está entre Durango y Coahuila. Este mercado es coloquialmente conocido como el “estado de la laguna”.



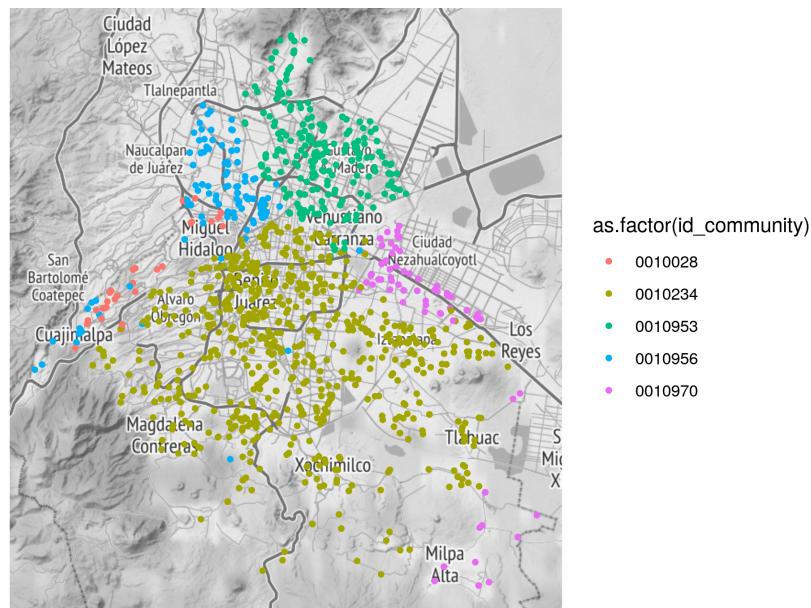
**Figura 4:** Mapa con las 15 comunidades más grandes

El 95 % de los mercados están dentro de un mismo estado. No obstante, cabe remarcar el caso del mercado en la intersección de Puebla, Tlaxcala el Estado de México e Hidalgo. Como se ve en la figura 5, el mercado está integrado por escuela provenientes de cuatro estados diferentes.



**Figura 5:** Mercado entre cuatro estados

Como se ve en la figura 6, en la Ciudad de México, existen cinco mercados diferentes

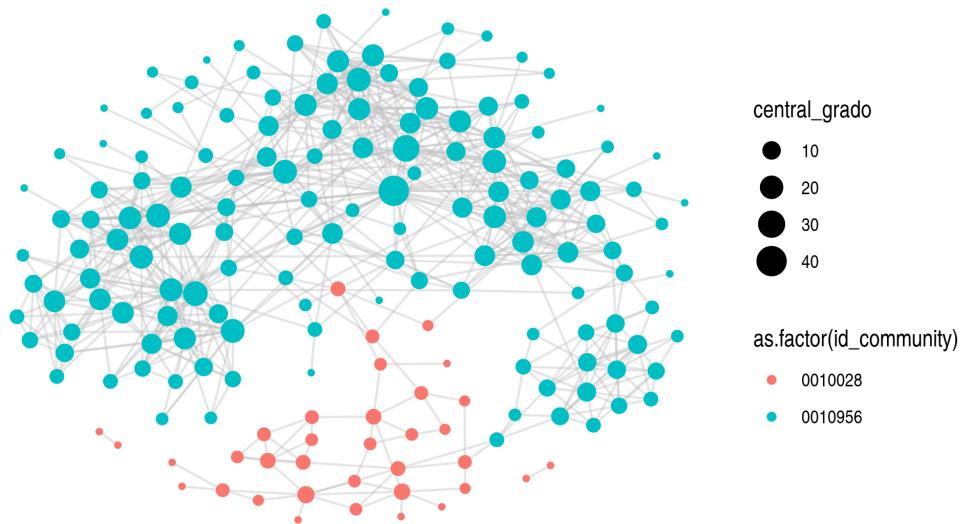


**Figura 6:** Mercados dentro de la Ciudad de México

En la figura 6, los mercados 0010956 y 0010028 geográficamente se traslanan. En la figura 7

se pueden visualizar las conexiones entre ambos mercados.

En la figura 7, el tamaño del nodo depende del grado. El grado es una medida de centralidad que cuenta cuántos aristas conectan con cada nodo. El grado de los nodos del mercado más grande es mayor que el grado del mercado más pequeño.



**Figura 7:** Grado de dos mercados en la Ciudad de México

En la figura 8, en cambio, el tamaño de los nodos corresponde a la intermediación varía más entre comunidades. Recordemos que la intermediación es la importancia de un nodo para conectar a otros pares de nodos en la red. Esto coincide con los puntos con intermediación alta que conectan ambas comunidades.



**Figura 8:** Intermediación de dos mercados en la Ciudad de México

## 9. Conclusiones y recomendaciones

En conclusión, se crearon los mercados educativos creando *commuting zones* con un radio de 10 kilómetros y dentro de cada *commuting zone* se probaron cinco algoritmos de detección de comunidades. Finalmente, los mercados se construyeron utilizando el algoritmo que maximizara la modularidad en cada *commuting zone*. Utilizando esta definición, se obtuvo una modularidad de 0.91 que superó al baseline de 0.76 en el cuál los mercados se construyeron utilizando únicamente los municipios de los mercados.

Como recomendaciones futuras, vale la pena utilizar características de la infraestructura y del personal de las escuelas para entender qué caracteriza a los mercados.

## Referencias

- [1] C. Maroy, "Why and how to regulate the education market?", *Profesorado, Revista de Currículum y Formación del Profesorado*, vol. 12, n.º 2, pág. 11, 2008.
- [2] C. Neilson, "Targeted vouchers, competition among schools, and the academic achievement of poor students", *Job Market Paper*, 2013.
- [3] L. M. Christian Posso Suárez, *School retention and crime, evidence from a nationwide education policy in Colombia*, Work in progress, 2020.
- [4] Y.-Y. Ahn, J. P. Bagrow y S. Lehmann, "Link communities reveal multiscale complexity in networks", *nature*, vol. 466, n.º 7307, págs. 761-764, 2010.
- [5] F. González, *Detección de comunidades*, Notas de clase Métodos analíticos, ITAM 2020, 10 de mayo de 2020.
- [6] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices", *Physical review E*, vol. 74, n.º 3, pág. 036104, 2006.
- [7] U. N. Raghavan, R. Albert y S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks", *Physical review E*, vol. 76, n.º 3, pág. 036106, 2007.
- [8] P. Pons y M. Latapy, "Computing communities in large networks using random walks", en *International symposium on computer and information sciences*, Springer, 2005, págs. 284-293.
- [9] M. Girvan y M. E. Newman, "Community structure in social and biological networks", *Proceedings of the national academy of sciences*, vol. 99, n.º 12, págs. 7821-7826, 2002.
- [10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte y E. Lefebvre, "Fast unfolding of communities in large networks", *Journal of statistical mechanics: theory and experiment*, vol. 2008, n.º 10, P10008, 2008.

# Appendices

## A. Código Switchers Primaria y secundaria

```
// Guarda para cada año el cct y curp
foreach anyo1 in 06 07 08 09 10 11 12 13 {
    local num = `anyo1'
    if `anyo1' == 06{
        local num = 6
    }
    if `anyo1' == 07{
        local num = 7
    }
    if `anyo1' == 08{
        local num = 8
    }
    if `anyo1' == 09{
        local num = 9
    }
    use "$basesAS/B`anyo1'.dta", clear
    keep cct curp grado
    duplicates tag curp, generate(dup_curp)
    keep if dup_curp == 0
    drop dup_curp
    gen anyo = `num'
    preserve
        keep if grado < 7 //primarias
        save "$basesD/B`num'_prim.dta",replace
    restore

    keep if grado > 6 //secundarias
    save "$basesD/B`num'_sec.dta",replace
}

foreach nivel in prim sec{
    // En año
    forvalues anyo1 = 6/12 {
        use "$basesD\B`anyo1'_`nivel'.dta", clear
        drop grado anyo
        local sig = `anyo1' + 1
        rename cct cct_origen
        merge 1:1 curp using "$basesD\B`sig'_`nivel'.dta", force
        rename cct cct_destino
```

```

preserve
    keep if _merge == 1
    drop _merge cct_destino
    save "$basesD/B`anyo1'_no_merg_`nivel'.dta",replace
restore

    keep if _merge == 3
    drop _merge
    save "$basesD/B`anyo1'_merg_`nivel'.dta",replace
}

// Busca en años no consecutivos
// Si lo encunetra lo guarda, si no lo busca en el que sigue
forvalues anyo1 = 6/10 {
    local sig = `anyo1' + 2
    forvalues anyo_sig = 8/12 {
        capture confirm "$basesD\B`anyo_sig'_`nivel'.dta"
        if _rc != 0{
            use "$basesD/B`anyo1'_no_merg_`nivel'.dta", clear
            merge 1:1 curp using \
"$basesD\B`anyo_sig'_`nivel'.dta", force
            rename cct cct_destino
            preserve
                keep if _merge == 1
                drop _merge cct_destino
                save "$basesD/B`anyo1'_no_merg_`nivel'.dta", \
replace
            restore

                keep if _merge == 3
                drop _merge
                save "$basesD/B`anyo1'_corr_merg_`anyo_sig'_`nivel'.dta", \
replace
        }
    }
}

forvalues anyo1 = 6/10 {
    use "$basesD/B`anyo1'_merg_`nivel'.dta",clear
    forvalues anyo_sig = 8/12 {
        capture confirm \
"$basesD/B`anyo1'_corr_merg_`anyo_sig'_`nivel'.dta"
        if _rc != 0{
            append using \
"$basesD/B`anyo1'_corr_merg_`anyo_sig'_`nivel'.dta"
        }
    }
}

```

```

        }
    }
    save "$basesD/B`anyo1'_completo_`nivel'.dta",replace
}

use "$basesD/B11_merg_`nivel'.dta",clear
append using "$basesD/B12_merg_`nivel'.dta"
forvalues anyo1 = 6/10 {
    append using "$basesD/B`anyo1'_completo_`nivel'.dta"

}
save "$basesD/completo_`nivel'.dta",replace
keep if cct_origen != cct_destino
save "$basesD/switchers_`nivel'.dta",replace

bysort cct_origen: egen numSalen = count(cct_origen)
drop curp
bysort cct_origen cct_destino: egen numDest = count(cct_destino)
gen porc_cambio = numDest/numSalen
duplicates drop cct_origen cct_destino, force
save "$basesA/agregado_`nivel'.dta", replace

}

```

## B. Construcción de grafo

```

library(dplyr)
library(igraph)

relations <- df %>%
    mutate(cct_d_u = ifelse(cct_d < cct_o, cct_d, cct_o)) %>%
    mutate(cct_o_u = ifelse(cct_d < cct_o, cct_o, cct_d)) %>%
    group_by(cct_d_u, cct_o_u) %>%
    dplyr::summarise(weight = sum(numDest)/sum(numSalen)) %>%
    ungroup() %>%
    rename(cct_d = cct_d_u, cct_o = cct_o_u) %>%
    left_join(df, by = c("cct_d", "cct_o"))

u_cct_o <- relations %>% distinct(cct_o, .keep_all = TRUE) %>%
    select(cct_o, latitud_o, longitud_o) %>%
    rename(lat = latitud_o, lon = longitud_o, name = cct_o)

u_cct_d <- relations %>% distinct(cct_d, .keep_all = TRUE) %>%

```

```

select(cct_d, latitud_d, longitud_d) %>%
  rename(lat = latitud_d, lon = longitud_d, name = cct_d)

nodos <- rbind(u_cct_d, u_cct_o) %>% distinct(name, .keep_all = TRUE)

school_network <- graph_from_data_frame(relations, directed=FALSE, vertices=nodos)
is_weighted(school_network)

```

## C. Construcción de commuting zones

```

create_buffers <- function(n_escuelas, buffer_r=1000, vec_lon=df_o$longitud, vec_lat=df_o$latitud)
  "Tras recibir un vector de longitudes y latitudes, toma aleatoriamente una muestra de observaciones y devuelve matriz con coordenadas y con el número buffer asociado. También deja en el entorno global la variable 'buffers_sf' que tiene la información espacial de los buffers calculados.

  * argumentos:
    ** n_escuelas: # escuelas que se toman al azar de la matriz de coordenadas
    ** buffer_r: radio del buffer a crear (en mts)
    ** vec_lon: vector con las longitudes de las escuelas
    ** vec_lat: vector con las latitudes de las escuelas

  * salidas:
    ** buff_mat: matriz con coordenadas y con el número buffer asociado"
  "

  set.seed(333814)
  # conformar matriz de coordenadas
  matrix_df <- data.frame(cbind(vec_lon, vec_lat))
  names_df <- data.frame(cct)
  draw <- sample(nrow(matrix_df), n_escuelas, replace = FALSE)
  mat <- matrix_df[draw, ] # aleatoriamente tomar n_escuelas escuelas
  names_cct <- names_df[draw, ]

  names(mat) <- c("longitud", "latitud")
  names(names_cct) <- c("names")

  mat_info <- mat # matriz que tendrá info de coordenadas y buffers
  # proyecciones
  unproj <- CRS("+proj=longlat +datum=WGS84") # proyección WGS84
  # proyectado a UTM para Mexico Ver: https://epsg.io/6370
  proj <- CRS("+init=epsg:6370")
  coordinates(mat) <- c(x="longitud", y="latitud") # convertir a shapefile
  proj4string(mat) <- unproj # asignar una proyección

```

```

# reproyectar el shapefile a WGS84 UTM 42N (para México).
mat <- spTransform(mat, proj)

# crear buffers (donde width está en mts, i.e. 1000=1km)
buffers <- gBuffer(mat, width=buffer_r)
buffers_sf <- st_as_sf(buffers)

# añadir columna a matriz para que diga de qué buffer es
mat_info$buff_num <- as.character(over(mat, disaggregate(buffers)))
print(paste("# buffers: ", length(unique(mat_info$buff_num))))
buff_mat <- mat_info

buff_mat$name <- names_cct

return(buff_mat)
}

```

## D. Comparación de algoritmos

```

library(dplyr)
library(igraph)

save_subgroups <- function(fc, select_nodos, algorithm){
  fccommunity<- membership(fc)

  select_nodos$sub_grupo <- 0
  tam <- nrow(select_nodos)
  for (i in 1:tam){
    escuela <- select_nodos$name[i]
    select_nodos$sub_grupo[i] <- fccommunity[escuela][[1]]
  }

  file_name <- str_c("./../../results/school_clusters/groups/select_nodos/",
                      algorithm, ".csv")
  sub_df <- select_nodos %>% select(name, sub_grupo)
  write_csv(sub_df, file_name)
  return(select_nodos)
}

compare_clustering_algorithms <- function(fc, select_nodos){
  algorithm <- str_replace(algorithm(fc), " ", "_")
  select_nodos <- save_subgroups(fc, select_nodos,

```

```

algorithm)

save_map(select_nodos,algorithm)
get_stats_group(select_nodos, algorithm)
return(get_community_stats(fc))
}

df1 <- compare_clustering_algorithms(cluster_walktrap(school_network),
                                      nodos)
df2 <- compare_clustering_algorithms(cluster_fast_greedy(school_network),
                                      nodos)
df3 <- compare_clustering_algorithms( cluster_label_prop(school_network),
                                      nodos)
df4 <- compare_clustering_algorithms(cluster_leading_eigen(school_network),
                                      nodos)
df5 <- compare_clustering_algorithms(cluster_louvain(school_network),
                                      nodos)
df_comparison <- rbind(df1,df2,df3,df4,df5)

```

## E. Cálculo de modularidad

```

members <- as.double(select_nodos$id_community)
nam <- as.character(select_nodos$name)
comms <- list(membership=members, vcount = vcount(school_network),
               name = nam,algorithm="by.hand" )
class(comms)<- "communities"
modularity(school_network, membership(comms))

```