Predicción de delitos y análisis en tiempo real del municipio de Zapopan con Twitter y Google

Datatón 2014

Equipo MSCGM

México

Luis M. Román García
ITAM

Omar Trejo Navarro
ITAM

Junio de 2014

1. Objetivo

Caracterizar la distribución de delitos en el municipio de Zapopan, y crear un sistema[1] que analice el *stream* de Twitter en tiempo real para encontrar sucesos clave y, utilizando un sistema de Google, brindar información relevante.

2. Predicción de delitos

Para estimar la densidad de delitos en Zapopan utilizamos una técnica no paramétrica similar a un histograma. Segmentamos el territorio que cubre Zapopan en celdas uniformes de $200m \times 200m$ (aproximadamente equivalente a una *manzana*) para hacer el análisis por zonas de extensión equivalente y utilizamos las georeferencias de la base de datos `Delitos.csv` para contar el número de delitos por celda.

2.1. Metodología

Se integraron varias bases de datos[2], donde cada una agrega varias variables que son utilizadas para predecir delitos.

Sabiendo el tipo de lugares y servicios que hay en cada celda y si han habido o no delitos se entreno el algoritmo *Stochastic Gradient Boosting*[3] para predecir futuras ocurrencias de delitos.

[1] Este proyecto se realizó utilizando R y el código se encuentra en https://github.com/Datata/Dataton.

[2] `Tianguis.csv`, `Eventos.csv`, `Servicios.csv` y `Delitos.csv`. Todas provistas en la página de la convocatoria http://dataton.datos.gob.mx/.

[3] Algunas de las ventajas se pueden consultar en [1].

Se utilizó el 70 % de los datos para entrenamiento del algoritmo y 30 % para la validación. Se utilizaron 7,501 muestras *bootstrap* para estimar el error de predicción. Se corrieron 83,600 árboles variando el grado de complejidad entre .1 y .01, y la profundidad de los cortes en $\{1, 3, 5, 7\}$. La máxima precisión se alcanzó con 700 árboles, 7 cortes y .1 de complejidad.

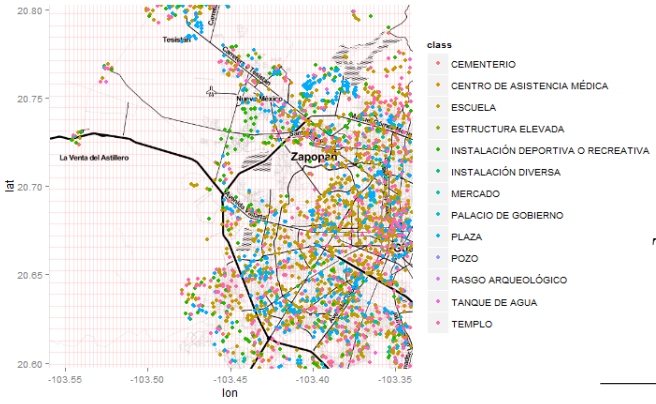


Figura 1: Distribución de lugares y servicios.

2.2. Resultados

La base de datos utilizada contiene delitos en 90 % de las celdas. Nuestro modelo consigue una clasificación correcta cerca del 97 % de los casos (considerablemente mayor a 90 %), lo cual es muy bueno.

Medida	Valor
Precisión	96.96 %
IC de 95 %	(96.21 %, 97.60 %)
Valor-p	$< 2e - 16$
Sensitividad	0.9845
Especificidad	0.8277

Tabla 1: Resultados del modelo óptimo.

	Referencia	
Predicción	Hubo delito	No hubo delito
Hubo delito	2226	41
No hubo delito	35	197

Tabla 2: Mátriz de confusión del modelo óptimo.

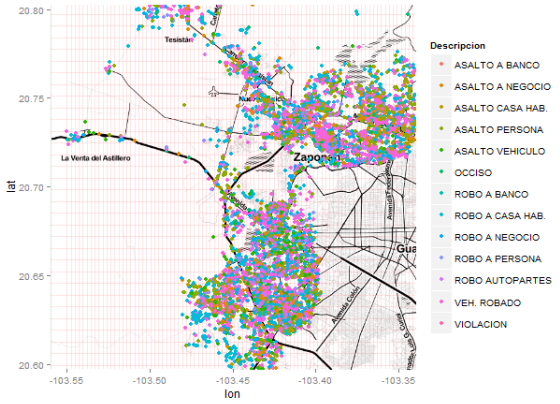


Figura 2: Distribución de crimen en Zapopan.

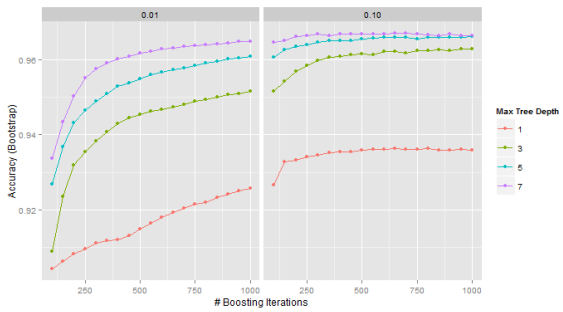


Figura 3: Presición vs. número de árboles.

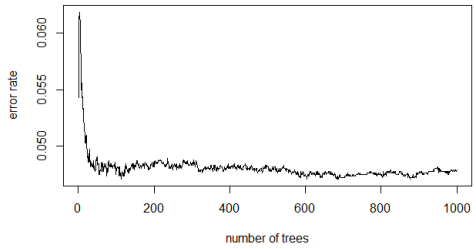


Figura 4: Tasa de error de predicción.

2.3. Análisis resultados

Como muestran los resultados, es posible asignar de manera precisa una medida de probabilidad de incidencia delictiva a cada segmento del territorio. Esto nos permite identificar las características que comparten las regiones con baja y alta incidencia delictiva. El siguiente paso es cruzar este análisis con datos obtenidos de redes sociales para optimizar el proceso detección y prevención del crimen.

3. Análisis en tiempo real con Twitter y Google

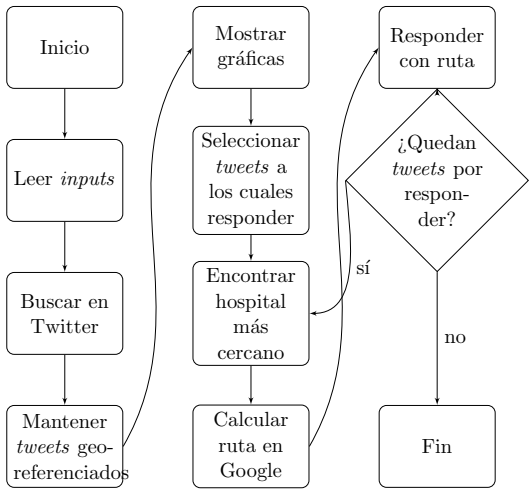
Utilizando la API REST de Twitter[4] y la API Directions de Google[5] logramos capturar *tweets* que informan sobre sucesos que ocurren en tiempo real, como robos, asesinatos, disparos, peleas, asaltos, entre otros, y generamos una respuesta con información relevante, en este caso el hospital más cercano, el tiempo que se tomará en llegar (tomando en cuenta el tráfico) y las instrucciones para hacerlo.

Existen varios herramientas para generar información útil en situaciones de crisis. Los más notorios son Google Crisis Response[6] y iSAR+[7]. De este último se puede consultar un análisis completo en [2]. El sistema que hemos diseñado en este proyecto se puede adaptar como un componente de sistemas como los que se acaban de mencionar.

3.1. Metodología

La estructura general del algoritmo se muestra en Fig. 5[8]. A continuación explicaremos cada uno de los nodos del diagrama.

Figura 5: Estructura general del algoritmo.



[4]Documentación en dev.twitter.com

[5]Documentación en developers.google.com/maps/documentation/directions/

[6]Más información en https://www.google.org/crisisresponse/.

[7]Más información en http://isar.i112.eu/

[8]Si se quiere analizar a un nivel de precisión mucho más profundo se puede consultar el código en https://github.com/Datata/Dataton. El código está bien documentado.

El **primer nodo** representa el incio del algoritmo. En el **segundo nodo** se leen los *inputs*. Leer *inputs* se refiere a leer el archivo de palabras clave (las palabras que se buscarán en Twitter), el centro de la zona geográfica que se quiere analizar (coordenadas: longitud y latitud) y el radio de la zona geográfica que se quiere analizar (en kilómetros), el número máximo de *tweets* que se quiere encontrar, y desde cuándo buscar *tweets* (sólo hoy, desde ayer, desde anteayer, etcétera).

En el **tercer nodo** se realiz la llamada al API de Twitter con el siguiente código:

```
tweets <- suppressWarnings(
    searchTwitter(key,
        geocode = geozone,
        since = since.date,
        n = number.tweets))
```

Utilizamos `suppressWarnings()` para no imprimir a pantalla cuando se encuentran menos *tweets* de los que se buscaban como máximo. La función `searchTwitter()` es parte de un paquete de R llamado *twitteR* que sirve para comunicarse con el API REST. Los parámetros `key`, `geozone`, `since.date` y `number.tweets`, son la palabra clave que se está buscando en ese momento, la zona geográfica donde queremos buscar, desde cuándo queremos *tweets* y el número máximo de *tweets* que queremos, respectivamente.

En el **cuarto nodo** lo que se hace es buscar dentro de todos los *tweets* recopilados aquellos que no contiene coordenadas (esto pasa porque al momento de buscar la API regresa aquellos *tweets* con geo-referncias o aquellos cuyas cuentas se pueden rastrear de forma inversa al lugar de la búsqueda), y los elimina porque aunque sepamos que la persona vive en esa zona, no sabemos el lugar exacto de donde se origina el *tweet*.

En el **quinto nodo** se muestra la gráfica con los resultados*Aquí podemos mostrar muchas otras estadísticas, como cantidad de* tweets *por persona, pero no son relevantes para este proyecto por lo que las omitimos.*. Un ejemplo de cómo se ve esta gráfica se encuentra en la Fig. 6.

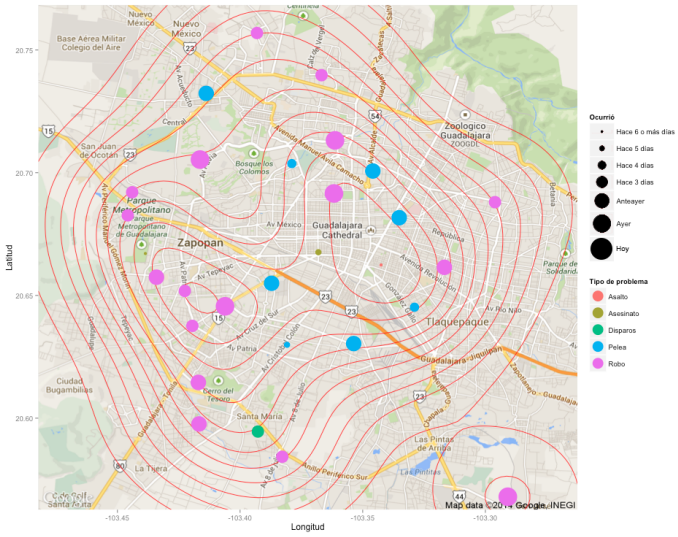


Figura 6: Resultados de *tweets* recientes en Zapopan con densidad estimada (lineas rojas).

En el **sexto nodo** se selecciona el algoritmo que identificará aquellos *tweets* que querramos analizar y a los cuáles generaremos una respuesta. En este caso el algoritmo es solamente buscar que contengan una de las palabras clave. Este paso se puede hacer mucho más robusto experimentando con otros algoritmos.

En el **séptimo nodo** buscamos el hospital más cerano al origen de algún *tweet* y lo hacemos utilizando la métrica Euclidiana. También aquí se puede experimentar con diferentes métricas si

queremos resultados con la máxima presición posible. El código para este paso es el siguiente:

```
entities$dist.to.location <-
    (as.numeric(location$Lng) -
        entities$Lng) ^ 2 +
    (as.numeric(location$Lat) -
        entities$Lat) ^ 2
min.dist.idx <- which.min(
    entities$dist.to.location)
nearest.entity <- data.frame(
    Lng = entities$Lng[
        min.dist.idx],
    Lat = entities$Lat[
        min.dist.idx])
```

Primero calculamos la distancia entre el origen del *tweet* y todos los hospitales (`entities$dist.to.location$`). Después buscamos la mínima de estas distancias y guardamos el índice (`min.dis.idx`). Finalmente, obtenemos las coordenadas del hospital más cercano (`nearest.entity`).

En el **octavo nodo** usamos las coordenadas del hospital más cercano para encontrar la mejor ruta con la API Directions de Google. El código para hacerlo es el siguiente:

```
 url <- paste("maps.google.com/
    maps/api/directions/json?",
    "origin=", from.coord,
    "&destination=", to.coord,
    "&language=", language.code,
    sep = "")
route <- fromJSON(paste(
    readLines(url),
    collapse = ""))
```

Esto obtiene una respuesta a través del API de Google en formato JSON[9] y lo transforma a una lista de R que podemos usar para extraer la información relevante. De aquí obtenemos la ruta óptima de un punto al otro[10], el tiempo aproximado que tomará en llegar la persona si fuese manejando (lo podemos modificar a que sea caminando) y la distancia en kilómetros.

Finalmente, en el **noveno nodo** se imprime la información obtenida a pantalla junto con el *tweet* que generó ese análisis, como se muestra en Fig. 7. Después, en el **décimo nodo** revisamos si ya terminamos de analizar todos los *tweets* que habíamos seleccionado en el sexto nodo; si no hemos terminado iteramos. Por último, el **décimo primer** nodo representa el fin del algoritmo.

3.2. Observaciones

Las observaciones que se presentan a continuación se enfocan al municipio de Zapopan. Lo primero que nos sorprendió es el hecho de que efectivamente hay personas que reportan incidentes graves como homicidos o robos con armas de fuego (Ej. *Homicidio en jimenez y valdez*), lo cual implica que nuestro proyecto tiene un impacto potencial enorme. Por otro lado, también nos dimos cuenta que mucha gente manda *tweets* que contienen palabras clave, pero las utilizan en contextos muy diferentes (Ej. *La mató con ese comentario*), lo cual representa ruído en el sistema y es un reto importante.

Con una inspección más profunda nos dimos cuenta que alrededor de 5 % de los *tweets* son avisos reales de sucesos de gravedad que acaban de ocurrir. Esto implica que para mejorar sustancialmente el algoritmo, y poder automatizarlo de manera importante, es necesario diseñar otros

[9] JavaScript Object Notation.

[10] Se puede crear una imagen con la ruta óptima y mandarla como archivo adjunto en la respuesta.

```
Tweet: Homicidio en jimenez y valdez

Palabra clave:   Homicidio
Distancia:       1,5 km
Tiempo:          5 min
Origen:          Roque Abarca 580-584, La Perla, 44360 Guadalajara, JAL, México
                 -103.335236, 20.681728
Destino:         Cruz Roja
                 Dr. Baeza Alzaga 91, Zona Centro, 44100 Guadalajara, JAL, México
                 -103.341915, 20.679400
Instrucciones:
                 1. Dirígete hacia el norte en Roque Abarca hacia Calle Pablo Valdez
                 2. Gira a la izquierda en la 2.ª bocacalle hacia Clemente Aguirre
                 3. Continúa por Calle Ignacio Herrera Y Cairo.
                 4. Gira a la izquierda hacia Calle Humbolt
                 5. Gira ligeramente a la izquierda para continuar en Calle Humbolt
                 6. Gira a la izquierda hacia Calle San Felipe
                 7. Calle San Felipe gira a la derecha hasta Dr. Baeza Alzaga
```

Figura 7: Resultado con instrucciones para el hospital más cercano.

componentes del algoritmo que detecten cuando una frase realmente implica algo que necesita atención y cuando no[11].

Dicho lo anterior, una vez que se ha identificado un *tweet* relevante, se genera una respuesta con la información en menos de cinco segundos, y se puede mandar inmediatamente, lo cual es un tiempo de respuesta excelente.

3.3. Generalización e impacto

Estos resultados son fácilmente generalizables. En menos de dos minutos, si se tienen datos de los hospitales en una base de datos, se puede utilizar el sistema para analizar cualquier zona del país (y del mundo) con tan sólo cambiar los parámetros iniciales del algoritmo.

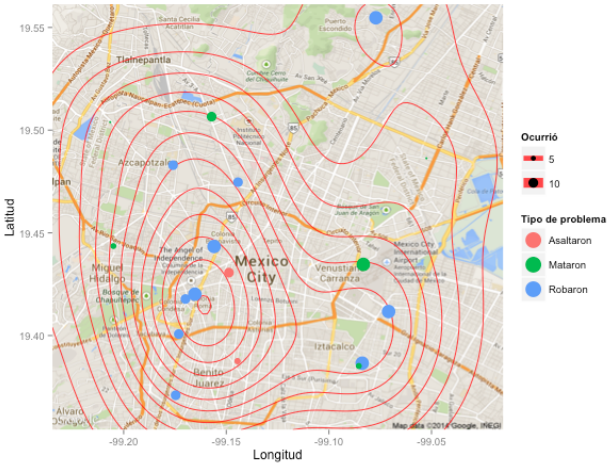


Figura 8: Resultados de *tweets* recientes en la Ciudad de México con densidad estimada (lineas rojas).

[11] Esto recae en el área de Natural Language Processing, una rama de Inteligencia Artificial.

Los resultados tardan en producirse alrededor de cinco segundos por cada *tweet* que se quiera analizar. Se muestran resultados de la Ciudad de México para mostrar la aplicabilidad del sistema en otras zonas geográficas.

Además de mandar la información al usuario, se se podría avisar al hospital del hecho a través de un correo electrónico monitoreado o un mensaje directo en Twitter a una cuenta para monitoreo de sucesos, y, además, podemos ir más allá y enviar patrullas o bomberos a lugares donde están ocurriendo estos acontecimientos (suponiendo que se cuenta con las coordenadas de las estaciones de policia o de bomberos para encontrar la más cercana y alguna manera de contactarlos virtualmente para hacerles saber que deben acudir a asistir). Esto reduciría drásticamente los tiempos de respuesta a sucesos que necesitan de acción inmediata.

Si iteramos este proceso, tenemos una forma automatizada de analizar muchos tipos de fenómenos sociales, económicos, naturales, y de cualquier otro tipo sobre el cual la gente publique en sus cuentas de Twitter, al mismo tiempo que podemos aprovecharla para responder con información o ayuda relevante que podría salvar vidas.

Referencias

[1] Brian Kriegler & Richard Berk, *Small area estimation of the homeless in Los Angeles: An application of cost-sensitive stochastic gradient boosting*, The Annals of Applied Statistics, Institude of Mathematical Statistics, Vol. 4, No. 3, 2010, pp. 1234-1255.

[2] Marco Manso & Bárbara Manso, *The role of social media in crisis: A European holistic approach to the adoption of online and mobile communications in crisis response and search and rescue efforts*, 17th ICCRTS, Operationalizing C2 Agility, iSAR+.