

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



SISTEMA DE RECOMENDACIÓN DE HOTELES  
SIMILARES

TESIS

QUE PARA OBTENER EL TÍTULO DE  
MAESTRO EN CIENCIA DE DATOS

PRESENTA

FELIPE GERARD VALDÉS

ASESOR: [Si me dejan] NADIA DEL VILLAR

MÉXICO, D.F.

2015

“Con fundamento en el artículo 21 y 27 de la Ley Federal del Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada “**SISTEMA DE RECOMENDACIÓN DE HOTELES SIMILARES**”, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la Biblioteca Raúl Bailleres Jr., autorización para que fijen la obra en cualquier medio, incluido el electrónico, y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por tal divulgación una contraprestación”

**Felipe Gerard Valdés**

---

FECHA

---

FIRMA

# Índice general

<b>Prólogo</b>	<b>VI</b>
<b>1. Análisis previo</b>	<b>5</b>
1.1. Sistemas actuales . . . . .	5
1.2. Diagnóstico . . . . .	11
<b>2. Sistema de recomendación de hoteles</b>	<b>14</b>
2.1. Modelo propuesto . . . . .	14
2.2. Precio . . . . .	16
2.3. Distancia: filtro estático . . . . .	17
2.4. Similitud de hoteles . . . . .	18
2.5. Distancia: filtro dinámico . . . . .	29
2.6. Precios dinámicos . . . . .	32
2.7. Consideraciones prácticas . . . . .	33
2.8. Juntando todo . . . . .	34
2.9. Comentarios . . . . .	34
<b>3. Implementación</b>	<b>35</b>
3.1. Información . . . . .	35
3.2. Sistema de respaldo . . . . .	37
3.3. Otras consideraciones . . . . .	38
3.4. Arquitectura final del sistema . . . . .	39
<b>4. Desempeño</b>	<b>41</b>
4.1. Análisis de redes . . . . .	41
4.2. Desempeño de la página . . . . .	47

<b>5. Conclusiones</b>	<b>52</b>
5.1. Trabajo futuro . . . . .	52
5.2. Comentarios finales . . . . .	52

# Índice de figuras

2.1.	Ejemplos de similitud de servicios para 4 categorías distintas. El hotel verde se parece al negro porque no le faltan muchos servicios que el negro sí tiene. En contraste, al hotel morado le faltan muchos servicios. . . . .	23
2.2.	Ejemplo univariado. Distancia entre $x$ y $y$ usando valor absoluto (---) y pérdida <i>hinge</i> (—). La pérdida <i>hinge</i> no penaliza el caso en que $y > x$ . . . . .	24
2.3.	Ejemplos de similitud de perfil para 4 categorías distintas. El hotel verde se parece al negro porque aunque le faltan servicios, tiene aproximadamente la misma proporción de sus servicios en cada categoría. El hotel azul tiene muchos servicios, pero tiene un enfoque distinto (en este caso es un hotel de súper lujo muy enfocado a playa, mientras que el negro no tiene playa). . . . .	25
2.4.	La elección de $\alpha$ determina la dirección en la que avanza la similitud total al variar la de servicios y la de perfil. . . . .	28
2.5.	El filtro debe ser más laxo cuando hay pocos hoteles similares cercanos (izquierda) y más estricto cuando hay más (derecha). . . . .	31
3.1.	Diagrama de flujo del sistema de recomendación. Amarillo: bases de datos. Rojo: Interacción humana. Verde: Procesos en <i>R</i> . Azul: Procesos en SQL. . . . .	40
4.1.	El sistema actual le da mucho peso a pocos hoteles y además no permite llegar de un grupo a otro. . . . .	44
4.2.	Red para diversos destinos: Ciudad de México (azul-morado), Cancún (turquesa), Playa del Carmen (verde), Riviera Maya (fucsia), Isla Mujeres (azul), Cozumel (naranja), Tulum (rojo) y Cuernavaca (verde-amarillo). El tamaño de los nodos está dado por el PageRank. . . . .	48

4.3. Hoteles en Cancún. La importancia (PageRank) está dada por el tamaño de los nodos. Sin contar los nodos inaccesibles, el diámetro de esta gráficaes de 7 pasos y la distancia promedio entre nodos es de 3.3. . . . .	49
4.4. Tamaño: PageRank. Color: Más oscuro implica mayor cercanía (en recomendaciones) al hotel La Quinta Inn and Suites. . . . .	50
4.5. Tamaño: PageRank. Color: Más oscuro implica mayor cercanía (en recomendaciones) al hotel Iberostar Cancún. . . . .	51

# Índice de tablas

# Prólogo

Las agencias de viajes han migrado la mayor parte de su negocio a las ventas en línea. Eso implica que junto con sus precios y la calidad de su servicio, la experiencia de usuario en sus páginas web es una parte crucial si quieren conseguir una ventaja competitiva. Dado que la mayor parte de las agencias empezaron antes de la era del Internet, el cambio al mundo digital ha sido paulatino y por necesidad, más que rápido y por gusto. Dado que el negocio consiste en *vender* viajes, la primera prioridad que se tiene al hacer una página es que sea operativa. La segunda prioridad normalmente es que sea agradable a la vista y aceptablemente fácil de usar. Estos dos principios tienen mucho sentido y de hecho los portales web actuales de las agencias de viajes los siguen razonablemente bien. Sin embargo, la era del Internet es mucho más que un punto de venta en cada casa.

Sería imprudente pensar que la experiencia de comprar en casa vía electrónica es exactamente igual que comprar en un quiosco de una agencia. La diferencia clave es el conocimiento que tienen los vendedores, que está ausente en una página web. La razón por la que esto es importante es que los clientes muchas veces no saben bien lo que quieren, y mucho menos lo que hay disponible. Entonces, suponiendo que se tiene resuelta la parte estática de la página (i.e. que sea funcional, utilizable y agradable), la tercera prioridad debería ser sustituir de algún modo al vendedor. Para ello es necesario hacer que la página sea dinámica y que tome ciertas decisiones automáticamente, de modo que pueda ayudarle al cliente a encontrar (y por lo tanto a comprar) la mejor opción disponible.

Usualmente, el primer componente de las páginas web que se busca que sea inteligente es el orden en el que aparecen los hoteles cuando se busca un destino en particular, también llamado *Sort Order*. La razón de esto es simple: dado que se sabe a dónde quiere ir el cliente, hay que darle las mejores opciones. Aquí el término “mejores” es, desde luego, desde el punto de vista

del cliente, pero en última instancia lo que se busca es aumentar las ventas y la rentabilidad de la empresa. Es común que en esta parte haya filtros adicionales, como el rango de precio y la categoría del hotel (es decir, sus estrellas) que permitan al cliente refinar su búsqueda. Si bien esto funciona bastante bien, también es cierto que en un destino con muchas opciones puede seguir siendo demasiado vago y, además, obliga al cliente a ser su propio asesor. La pregunta ahora es qué más se puede hacer para incorporar información y así ayudar a cerrar ventas.

Conocer el destino del viaje del cliente es cercano a lo mínimo que se puede saber sobre él. Sin embargo, cuando se mete a la página de un hotel en particular nos proporciona bastante más información. En particular, nos da una idea de su presupuesto, de la categoría que espera que tenga su hotel y también algo de información geográfica más detallada que sólo una ciudad o zona. A menos que apliquemos un cuestionario, no tenemos otra manera de obtener más datos sobre lo que el cliente quiere, así que a lo más que podemos aspirar es a recomendar hoteles similares al que el cliente vio en primera instancia. La esperanza aquí es que si el cliente casi encontró lo que quería en la primera opción, probablemente le guste algún hotel similar cercano. Si aún así no compra, queremos que siga y siga viendo alternativas hasta que encuentre una lo suficientemente buena. Un sistema con estas características podría jugar el papel de asesor y, de hecho, los vendedores humanos hacen algo muy similar: escuchan al cliente y proponen una opción, reciben retroalimentación y vuelven a proponer hasta que se cierra una venta o el cliente se retira.

Ahora bien, para que el sistema que hemos delineado tenga éxito, es necesario que tenga varias características. Entraremos en detalles a lo largo de la tesis, pero a hay ciertas propiedades básicas que deberá tener si planea sustituir a un vendedor:

- Deberá *pensar lo más parecido a un humano*, es decir, tener un concepto de similitud y cercanía entre dos hoteles lo más parecido posible al de una persona.
- Deberá ser *inteligente y flexible* al recomendar. Es importante que proponga alternativas viables sin caer en repeticiones. Imagine el lector que un vendedor insiste en una opción que el cliente no aprueba.
- Deberá *ser empático y atento*, o también, tomar en cuenta el presupuesto y los cambios que el cliente proponga (por ejemplo, si el cliente

repentinamente está más interesado en hoteles más económicos que antes, se deberán ajustar las recomendaciones).

- Deberá ser *ingenioso* cuando las circunstancias sean difíciles y recomendar siempre algo, aún cuando haya poco de dónde escoger.

Aunque el lector podría pensar que los adjetivos utilizados en los puntos anteriores son ambiciosos, en realidad corresponden a funciones muy básicas y perfectamente evaluables en el contexto que nos interesa. Esperamos que al final de esta lectura lo habremos convencido de que nuestro sistema en efecto es capaz de sustituir razonablemente bien a un vendedor.

En el presente trabajo desarrollaremos paso a paso un *modelo de recomendación de hoteles similares* con la filosofía descrita anteriormente, hecha a la medida especialmente para las necesidades de la empresa Best Day Travel (de aquí en adelante sólo “Best Day”). La inquietud que tuvo esta empresa por mejorar la sección de “Hoteles Similares” en su página fue la que llevó a la concepción de este proyecto, por lo que el objetivo siempre fue no sólo mejorar con respecto a lo que se tenía, sino también con respecto a la competencia. Debido a la naturaleza tan aplicada del proyecto, incluimos un capítulo entero dedicado a temas técnicos relacionados con la implementación particular en el sistema actual de Best Day. A continuación describimos brevemente la estructura de la tesis.

En el primer capítulo (1) describiremos el *status quo* tanto de la competencia como de Best Day con el fin de aprender de lo que ya hay y así poder mejorarlo. Estos análisis nos resultaron muy útiles para establecer los objetivos de nuestro sistema.

En el segundo capítulo (2) describiremos a detalle el modelo que construimos, incluyendo las propiedades que deberá tener para que se comporte como deseamos y también qué medidas tomamos para solucionar cada una de las dificultades.

El tercer capítulo (3) tiene información técnica necesaria para la implementación del modelo en el escenario real de Best Day. Incluye el tratamiento de la información y las acciones a tomar en presencia de datos de mala calidad o faltantes. En pocas palabras, es la liga entre el capítulo 2 y el mundo real.

En el capítulo cuatro (4) hablaremos del desempeño del modelo, desde dos puntos de vista. El primero es desde el punto de vista del negocio, es decir cómo se comporta la página antes y después de implementar el modelo

propuesto. El segundo es desde el punto de vista del cliente, es decir si las recomendaciones tienen sentido y por lo tanto si hemos logrado hacer el trabajo un asesor humano. En esta segunda parte también incluiremos un análisis de importancia de los hoteles, que resulta naturalmente del modelo que elegimos.

Finalmente, concluiremos en el quinto capítulo (5), donde discutiremos algunos puntos importantes sobre el sistema y sus implicaciones y algunas ideas para trabajo futuro.

# Capítulo 1

## Análisis previo

Antes de desarrollar el modelo de recomendación de hoteles similares, es conveniente analizar el *status quo*. Comenzaremos con un análisis de la competencia, con el que extraeremos información valiosa de lo que hay actualmente en el mercado, y posteriormente haremos un análisis del sistema de Best Day. El objetivo del primero es aprender de las fortalezas y debilidades de otros para enfocar nuestro trabajo de la manera más eficiente posible, mientras que el del segundo es poner a prueba a nuestro “vendedor” electrónico actual y así ver qué le podemos enseñar.

### 1.1. Sistemas actuales

Para llevar a cabo tanto los análisis de la competencia como el propio es necesario establecer qué ámbitos de los sistemas de recomendación nos interesan. Dado que el presente trabajo está enfocado en la inteligencia detrás de las elecciones y no en un aspecto estético, nos enfocaremos en los siguientes dos campos:

1. La posición dentro de la página de las recomendaciones.
2. La forma en la que se eligen las recomendaciones.

Enfatizaremos en los principales competidores de Best Day, es decir: Booking, Price Travel, Expedia y Despegar. También veremos si podemos aprender algo de TripAdvisor, que aunque no es un competidor directo, vive de recomendar hoteles y de su comunidad de comentarios. Las empresas competidoras

están ordenadas según nuestra evaluación de la calidad de sus sistemas de recomendación. Las ordenamos de menos a más relevantes.

## **Price Travel (pricetravel.com)**

Pusimos a Price Travel en el primer lugar de la lista porque no tiene un sistema de recomendación de hoteles similares. Cuando se busca un destino, se presentan los hoteles en el Sort Order determinado por su sistema, pero ya que se entra a la página de un hotel en particular (independientemente de cómo se llegó a esa página), no se recomienda hoteles similares ni cercanos. De hecho, la única indicación de otros hoteles es en la pestaña de mapa, en la que se pueden ver todos los cercanos en un mapa interactivo. Ni siquiera intentaremos hacer el símil de este sistema con un asesor electrónico.

## **Booking (booking.com)**

Booking es un competidor muy importante de Best Day, aunque también tiene un mercado más grande y mayor cantidad de opciones en destinos fuera de Latinoamérica. Antes de entrar en los puntos de interés general, cabe mencionar que algo que hace booking que no hacen los demás es que cuando se busca un hotel en la página con la barra inteligente, en lugar de ir directamente a su página, se muestra en el primer lugar de una lista, dando la posibilidad de elegir algún otro hotel. Ahora sí pasemos al tema en cuestión:

### **▪ Hoteles populares por destino**

1. Se encuentran del lado izquierdo, abajo de la barra de búsqueda y fechas. Es una posición de prominencia media-baja, puesto que no es visible inmediatamente sin bajar en la página.
2. Cuando se entra a la página (no al hotel en particular) por primera vez y se selecciona un hotel, aparece una sección de hoteles populares en el mismo destino del hotel buscado. El tema es que siempre aparece la misma selección para cada destino, independientemente del hotel que se haya elegido, pues la popularidad de los otros hoteles no tiene relación con el hotel elegido.

### **▪ Búsquedas recientes**

1. Se encuentran en el mismo lugar que los hoteles populares por destino, pero los sustituyen una vez que se ha visitado al menos un hotel anteriormente.
2. La información de los hoteles visitados se va almacenando en una *cookie*, de modo que para los hoteles subsecuentes, en lugar de los hoteles populares por destino, aparece un listado de unos pocos hoteles recientemente buscados por el usuario.

Al parecer en Booking sólo están interesados en lo que el cliente ha buscado y utilizan los hoteles populares por destino para no dejar vacía la sección para los clientes nuevos. El problema con los hoteles populares por destino es que no varían según el hotel buscado (siempre y cuando estén en el mismo destino), mientras que el de las búsquedas recientes es que no promueven que el cliente explore la página y encuentre la mejor opción para sus necesidades. Podríamos comparar este sistema con un vendedor que propone una lista de hoteles primero y luego únicamente dice “a usted le habían gustado los hoteles *x, y y z*”. Sin duda esto es mejor que nada, pero veremos más adelante que se puede hacer bastante más.

## **Despegar ([despegar.com](http://despegar.com))**

Despegar es un competidor más directo y con un mercado más parecido al de Best Day que Booking, aunque está más enfocado en vender boletos de avión que cuartos de hotel. En su página utilizan varias estrategias distintas para recomendar, por lo que las describiremos por separado. Cabe mencionar que no usan todas las estrategias para todos los hoteles, ni siempre, por lo que sospechamos que tienen información incompleta y sólo muestran lo que tienen disponible.

### **■ Búsquedas recientes**

1. Se encuentra en la izquierda abajo del mapa. Igual que en [booking.com](http://booking.com), se tiene que bajar en la página para verlo, por lo que su relevancia es media-baja.
2. Son los hoteles buscados recientemente por el usuario. Curiosamente, no siempre nos aparecieron al hacer búsquedas, incluso después de haber visto varios hoteles.

### **■ Hoteles cercanos**

1. Se encuentra directamente debajo de las búsquedas recientes.
2. Se listan los hoteles más cercanos al hotel en buscado, sin ningún criterio de precio, categoría o estilo del hotel.

- **Hoteles populares por destino**

1. Se encuentra directamente abajo de los hoteles cercanos.
2. Para cada destino muestra una lista de los hoteles más populares (presumiblemente los más vendidos). La lista es la misma para todos los hoteles en un destino dado y de hecho contiene al hotel buscado en su posición correspondiente (es decir ni se omite ni aparece hasta arriba).

- **Otros usuarios también buscaron...**

1. Aparece en la parte inferior de la página, abajo de prácticamente todo. Su relevancia es muy baja y de hecho se omitió en los primeros análisis de la competencia porque no lo vimos.
2. En la parte inferior de la página también aparece una sección de hoteles que fueron visitados por los demás usuarios que han visto el hotel en cuestión. Este análisis no es complejo, pero puede ser pesado computacionalmente si no se utiliza un método apropiado.

En Despegar tuvieron muchas ideas para recomendar, pero todos los criterios están separados y no colaboran entre sí. Es como tener una tabla ordenada por varios campos distintos. Los hoteles recomendados con base en lo que han visto otros clientes creemos que están desperdiciados en donde están, puesto que sólo los clientes que vayan hasta abajo de la página los verán. El sistema funciona razonablemente bien y permite movilidad hacia arriba en la categoría del hotel. El problema es que es claramente sesgado a recomendar hoteles populares, de modo que tiende a atorarse en aglomerados de hoteles populares y también a subir de categoría. El resultado es que después de algunos pocos *clicks*, casi invariablemente se llega a los hoteles más caros de la zona, sin posibilidad de explorar otros.

La ventaja de Despegar es que al menos toman en cuenta que hay diversos criterios de interés. El problema es que su “vendedor”, o es incapaz de integrar varios criterios, o bien insistente, poco creativo y además nos quiere vender lo más caro que se pueda. Sin duda es mejor que el de Booking, pero aún podría mejorar bastante.

## **Expedia (expedia.com)**

En Expedia se le da una prioridad bajísima a las recomendaciones, puesto que están posicionadas en un lugar casi imposible de ver. Sin embargo, tienen un sistema diferente a los demás y por eso lo pusimos en esta posición en la lista de sistemas competidores.

### ▪ **Búsquedas recientes**

1. Se encuentran en el centro, casi hasta abajo, después de toda la información del hotel. La prioridad es muy baja.
2. Es igual que en las demás páginas. Aparecen los últimos cuatro hoteles buscados distintos del actual (en caso de repetir una búsqueda).

### ▪ **Hoteles recomendados**

1. Se encuentran directamente abajo de las búsquedas recientes.
2. Utilizan un método desconocido para recomendar hoteles similares que permite movilidad progresiva entre categorías. No estamos seguros si tiene alguna restricción de precios.

El sistema de recomendación de Expedia nos gustó bastante y de hecho creemos que está siendo desaprovechado. Permite explorar los hoteles disponibles hacia arriba y hacia abajo en la escala de precio y categoría y llegar a opciones completamente nuevas. Una ventaja sobre el algoritmo de Booking es que no tiene una tendencia tan fuerte hacia los hoteles caros. En cuanto a las desventajas, es probable que esté acotado por destino, aunque eso no es tan grave. Otro detalle que no nos convenció es que le falta algo de flexibilidad: después de seleccionar algunas pocas recomendaciones, es evidente que nunca saldremos de un cierto grupo de hoteles.

En conclusión, creemos que el enfoque que tomó Expedia es el correcto, puesto que emulan un vendedor que, por un lado, propone nuevas alternativas y, por otro, se “acuerda” de lo que ya le había gustado al cliente. Lo que le faltaría es ser menos tímido (i.e. poner las recomendaciones en un lugar de mayor impacto) y menos conservador (i.e. proponer opciones que se salgan de un círculo reducido de hoteles).

## Trip Advisor ([tripadvisor.com](http://tripadvisor.com))

Como decíamos arriba, Trip Advisor no es la competencia directa de Best Day, pero es un sitio muy popular de evaluación de hoteles y creemos que podemos aprender de ellos.

### ▪ Hoteles similares

1. Se encuentran en la derecha, abajo de la foto principal, pero al lado de otra información importante, por lo que el impacto es medio.
2. No sabemos cómo hacen las recomendaciones, pero no son por distancia estricta y sí permiten cierta movilidad de la categoría.

El sistema de TripAdvisor también funciona razonablemente bien, pero parece estar más basado en la calificación de los usuarios que en la categoría del hotel. Nos pareció que podría mejorar si tuviera una mayor flexibilidad, pues comparte el problema de la falta de inventiva del sistema de Expedia. A pesar de las faltas, este enfoque tiene sentido en el contexto de su página y funciona más o menos bien.

Es posible que el sistema de TripAdvisor sea un filtro colaborativo (ver [7] para mayor información) desde el punto de vista de los productos (i.e. *item-based*). Un sistema de este estilo podría funcionar bien si se tuviera clientes recurrentes, puesto que está basado en hacer silogismos del tipo “a los clientes que les gustó A también les gustó B”. Sin embargo, Best Day no tiene una recurrencia tan alta y tampoco tiene información suficientemente detallada de los clientes como para tratar de seguir por este camino. Por lo tanto, sería inviable implementar un sistema de este estilo.

## Best Day ([bestday.com.mx](http://bestday.com.mx))

Finalmente, analizaremos lo que se hace actualmente en Best Day. Del mismo modo que con la competencia, comenzaremos con una breve descripción:

### ▪ Hoteles similares

1. Los hoteles similares están abajo de la liga al mapa del lado derecho. Como en la mayoría de los competidores, no se ve inmediatamente al entrar en la página, por lo que tiene prominencia media-baja.

2. La manera de recomendar hoteles es por categoría (estrellas) y por destino, es decir, se toman los hoteles de las mismas estrellas y en el mismo destino que el hotel buscado y se presentan en el Sort Order del momento. No se toma en cuenta el precio ni la distancia (salvo por el hecho de encontrarse en el mismo destino) de las opciones.

#### ■ Búsquedas recientes

1. Se encuentran directamente debajo de los hoteles similares.
2. Similarmente a la competencia, aparece una lista de los hoteles que el usuario visitó recientemente. Las búsquedas recientes permanecen entre sesiones dentro de una misma computadora.

Sin duda, el hecho de tener una sección de búsquedas recientes es un punto a favor de este sistema de recomendación, aunque no aporta nada que no tengan los competidores. Desafortunadamente, la calidad de sus recomendaciones es muy variable. En zonas homogéneas y con mucha oferta para una categoría dada, las recomendaciones pueden ser aceptables. Sin embargo, en destinos donde hay hoteles muy distintos pero con la misma categoría, como podrían ser un hotel citadino y uno de playa, las recomendaciones dejan mucho que desear. Además, si no hay hoteles bajo este criterio, o si hay muy pocos, no se muestra nada, o una lista muy pequeña. Es decir, el “asesor” electrónico de Best Day maneja las recomendaciones en grupos: dada información del cliente (destino y estrellas) es sumamente insistente con las alternativas. En pocas palabras, no tiene ninguna ventaja sobre un catálogo agrupado por destino y estrellas.

## 1.2. Diagnóstico

En los siguientes párrafos resumiremos lo que aprendimos del análisis de la competencia y del estado actual de Best Day. En primer lugar, nos dimos cuenta de que poner una sección con las búsquedas recientes es algo muy deseable y relativamente fácil de implementar. De hecho, es un estándar en la industria, pues todos los competidores lo tienen, a excepción de Price Travel. Sin duda es deseable que un vendedor recuerde los artículos que le ha mostrado al cliente, aunque esto de ninguna manera es suficiente para crear una sensación de inteligencia.

Quitando las búsquedas recientes de en medio, podemos analizar los diversos sistemas de recomendación. Un punto que pensamos que es sumamente importante es el precio. Hasta donde pudimos ver, ninguna página tiene miramiento alguno con el costo de los hoteles recomendados. El resultado es que pueden recomendar hoteles del doble o incluso triple de precio que el original. Creemos que esto no promueve que el cliente explore más alternativas, ya que la mayoría basa su decisión fuertemente en el precio de los hoteles. Para ver que tomar el precio en cuenta es importante, basta imaginar que un vendedor en una tienda nos intenta vender artículos más y más caros. Sin duda es molesto y menos agradable que uno que busca encontrar productos que satisfagan nuestras necesidades sujeto a nuestras restricciones presupuestales. Propondremos una forma simple de lidiar con este problema sin perder demasiada flexibilidad en las recomendaciones.

Otro factor que nos pareció mejorable es el uso de la información geográfica. Parece ser que la opción más popular es únicamente utilizar la información incluida en el destino y, de hecho, parece que sólo Despegar tiene una sección de hoteles más cercanos. Si bien utilizar el destino como punto de partida es una opción razonable, creemos que se podría aprovechar mejor la información si se utilizaran las coordenadas geográficas, dado que en la vida real lo que realmente importa es la proximidad de los hoteles; los destinos son particiones más o menos arbitrarias que no deberían afectar mucho las recomendaciones. Por ejemplo, una persona podría recomendar un hotel en Cancún dado que se buscó uno en la zona norte de la Riviera Maya, o uno en Playa del Carmen a una persona que buscó algo en Cozumel.

Un tercer problema con el que nos encontramos en las recomendaciones es que en la mayoría de los casos eran estáticas. Es decir, son idénticas para ciertos grupos de hoteles. Esto pasa en todas las opciones que vimos que utilizan el destino (Booking, Despegar, Best Day) como criterio, pues no dependen también del hotel. Esto en sí mismo puede ser un problema en destinos amplios, puesto que el usuario agota las opciones rápidamente. Hay que decir que de ninguna manera decimos que las recomendaciones estáticas son malas, pero su función debe ser desempeñada por catálogos o por búsquedas avanzadas, no por sistemas de recomendación

En cuarto lugar, nos dimos cuenta que en varias de las páginas se han dado cuenta de que quieren recomendar hoteles vía diversos criterios, pero muchas veces no los han integrado. La estrategia tiende a ser usar un criterio para ordenar a la vez, en lugar de aprovechar varios simultáneamente. Este punto va de la mano con el tercero: los criterios estáticos deben ir en

una búsqueda avanzada. Es necesario integrarlos de alguna forma para que parezcan, aunque sea de lejos, inteligentes.

En quinto lugar, notamos que en general los sistemas que sí recomendaban de forma más sofisticada resultaban rígidos y se podían quedar atorados fácilmente en grupos aislados de hoteles. Esto hace que el sistema se vuelva inútil una vez que se llegó a uno de estos grupos absorbentes. El único que nos pareció decente en este sentido es el de Expedia, aunque sí es algo más rígido de lo que nos gustaría. No podemos dejar de enfatizar la importancia que tiene la “creatividad” en este tipo de sistemas. De igual modo que los vendedores reales, el valor que aportan reside en sus ideas, su inventiva y sus propuestas innovadoras. Es por ello que si se les agotan las ideas (o en este caso, si se ciclan en un grupo de ellas que no sea lo suficientemente bueno) se vuelven obsoletos.

Otro punto sobre los sistemas en general, pero de los de hoteles similares en particular, es que en ninguna página (salvo tal vez Trip Advisor) aparecen en posiciones de importancia alta. En el mejor de los casos aparecen una página abajo de la principal, al lado, y en los peores casos están hasta abajo. Creemos que se está desaprovechando mucho el potencial de dichos sistemas. Es un hecho que los vendedores tímidos no son tan efectivos como podrían. En este caso es posible hacer que todos los “vendedores” se vuelvan instantáneamente tan extrovertidos como se deseé; creemos que se debería aprovechar esta capacidad. No se trata de hacerlos intrusivos ni exagerados, pero sí de darles las herramientas (la exposición) que requieren para hacer mejor su trabajo.

Finalmente, un problema particular del sistema de Best Day es que no toma en cuenta de ninguna forma el perfil del hotel. Como mencionamos arriba, no es lo mismo un hotel de playa de cuatro estrellas que un hotel de ciudad de cuatro estrellas, y si por alguna razón ambos están en el mismo destino, como puede suceder en ciudades cerca del mar, por ejemplo, el sistema de recomendación por destino y estrellas puede fallar catastróficamente. Tenemos ejemplos de hoteles de negocios en Cancún para los que se recomienda hoteles todo incluido en la zona hotelera. Esto no es viable y claramente es importante que el algoritmo que desarrollemos tome en cuenta el perfil de los hoteles, sobre todo porque busca recomendar hoteles *similares* de una manera similar a lo que haría un experto.

# Capítulo 2

## Sistema de recomendación de hoteles

En el capítulo anterior analizamos los sistemas de recomendación que tienen actualmente tanto Best Day como su competencia y expusimos los principales puntos débiles que encontramos. Ahora podemos utilizar el conocimiento obtenido para generar un sistema nuevo que tenga un mejor desempeño y propiedades más deseables que los que vimos. Primero describiremos punto por punto qué nos gustaría que hiciera el sistema y posteriormente dedicaremos una sección para profundizar en cómo resolvimos cada uno de los problemas.

### 2.1. Modelo propuesto

Proponemos las siguientes medidas con el fin de propiciar que las recomendaciones tengan un carácter más parecido al de un verdadero vendedor:

1. Con el fin de que sea *empático*, pondremos una cota superior al **precio** de los hoteles recomendados, como un porcentaje del precio del hotel buscado.
2. Para que tome en cuenta la distancia *lo más parecido posible a una persona*, proponemos que recomiende hoteles razonablemente **cercanos** al buscado, donde el criterio de cercanía varía según la densidad de hoteles en la zona.

3. Para que *compare hoteles como un vendedor*, las recomendaciones deberán tener un **perfil** o filosofía similar al original. Además, las recomendaciones deben ser hoteles no necesariamente de las mismas estrellas, pero sí deben tener una cantidad similar de **servicios** disponibles.

En la lista anterior aparentemente hemos dejado fuera las características de *ingenio* y de *flexibilidad*. Sin embargo, más adelante en este capítulo y en el capítulo 3 veremos qué hicimos para tratar con casos atípicos o con muy poca población. Por otro lado, en el capítulo 4 quedará claro que el sistema propuesto naturalmente genera una estructura diversa que difícilmente se quedará enfascada en unas pocas opciones

Antes de entrar a una descripción detallada de cada medida, explicaremos brevemente cómo colaborarán a que nuestro sistema de recomendación tenga las propiedades que buscamos.

Siempre y cuando dos hoteles sean genuinamente similares a los ojos de una persona, la vasta mayoría de los clientes siempre preferirá el más económico. Acotando el precio esperamos que rara vez aparezcan hoteles más allá de las posibilidades del cliente en las recomendaciones. Esto contribuirá a que el sistema sea *empático* porque tendrá en cuenta el presupuesto del cliente y se ajustará conforme éste explore alternativas de precios distintos.

En cuanto a la distancia entre hoteles, creemos que utilizar la distancia al hotel en cuestión es mucho más razonable que usar el mismo criterio del destino para todos, puesto que promoverá hoteles cercanos además de parecidos en otros sentidos. Dado un hotel en un mapa, una *persona* naturalmente se fijaría en la distancia en línea recta a las alternativas, no utilizaría las fronteras arbitrarias de los destinos.

Esperaríamos que las similitudes de perfil y servicios trabajen conjuntamente con el criterio de cercanía para obligar a que las recomendaciones varíen de hotel a hotel, incluso dentro de un mismo destino. La ventaja de esto es que se puede explorar muchas opciones sin tener que buscar nada explícitamente, puesto que las recomendaciones abren nuevas alternativas por sí solas. Es importante que demos suficiente flexibilidad al método para que no sea rígido y no se estanque en grupos reducidos de hoteles. Estos dos criterios en su conjunto contribuirán tanto a que el “asesor” *compare* como una persona, como a que sea *creativo*.

Como último punto, no queremos utilizar estos criterios de forma aislada, pues cada uno tiene diversas faltas. Lo que queremos es integrarlos todos en un sistema poderoso que haga las veces del vendedor, es decir, que evite que

el cliente tenga que pensar demasiado y que genuinamente le proponga alternativas interesantes. Para ello generaremos una medida de similitud entre los hoteles y mostraremos los más parecidos que estén a una distancia y precio razonables del hotel original. En las siguientes secciones describiremos cada uno de los criterios con mayor nivel de detalle.

Para facilitar la exposición le daremos nombres a los hoteles que van superando cada etapa, hasta llegar a las recomendaciones para cada hotel.

**Definición 2.1** (Conjunto de hoteles). *Denotamos el conjunto de todos los hoteles por  $\mathcal{H}$  y al número de hoteles por  $N$ , es decir  $N = |\mathcal{H}|$ .*

## 2.2. Precio

Dado que el precio es un factor determinante en la elección de un hotel para la mayoría de los clientes, es importante respetar sus posibilidades económicas a la hora de recomendar nuevas opciones. Es razonable pensar que dados dos hoteles idénticos, un cliente siempre elegirá el más barato, salvo tal vez unos pocos clientes de súper lujo. Por esta razón, decidimos no imponer ninguna cota inferior al precio de los hoteles recomendados, siempre y cuando cumplan los criterios de similitud que describiremos más adelante. Por otro lado, como puede ser desalentador que las sugerencias contengan opciones demasiado caras, optamos por poner una cota superior al precio de los hoteles. Como en este paso queremos reflejar los precios “reales” de los hoteles, utilizaremos su precio de largo plazo para generar el filtro, como se explica a continuación.

**R 1** (Filtro de precio de largo plazo). *Sea  $\gamma \geq 0$  y sea*

$$p_i^D = \frac{\text{Importe total del hotel } i \text{ en el periodo } D}{\text{Total de noches del hotel } i \text{ en el periodo } D}$$

*el precio de largo plazo del hotel  $i$  en el periodo  $D$ . Entonces definimos la cota superior de precio para  $i$  por  $(1 + \gamma)p_i^D$ , es decir, excluimos del análisis para  $i$  todos los hoteles  $j$  con  $p_j^D > (1 + \gamma)p_i^D$ . Los hoteles restantes están contenidos en el conjunto*

$$\mathcal{H}_i^1 = \{j \in \mathcal{H} : p_j^D \leq (1 + \gamma)p_i^D\}$$

El parámetro  $\gamma$  es esencialmente el porcentaje por encima del precio de  $i$  que se permitirá que tengan las recomendaciones. Por ejemplo, si  $\gamma = 0,3$ , entonces se ignoran todas las opciones que sean más de 30 % más caras que  $i$ . El periodo  $D$  debe ser suficientemente largo para que refleje los precios promedio de un hotel, pero no tanto como para que se vea demasiado afectado por la inflación y otros factores. Una buena elección es por ejemplo un año o dos años, pues así además se toma en cuenta los cambios de precios en las diversas temporadas del año. Cabe notar que en esta versión del modelo, tanto  $\gamma$  como  $D$  son los mismos para todo hotel  $i$ .

En pruebas empíricas nos gustó el desempeño de  $\gamma$  alrededor de 0.3, ya que permite suficiente flexibilidad en los precios y las categorías sin proponer opciones excesivamente costosas.

### 2.3. Distancia: filtro estático

Antes de entrar en criterios dinámicos de distancia y en similitudes de hoteles, es importante pensar si su cálculo es factible. El tema es que, si bien en teoría sería suficiente calcular las similitudes de los hoteles y luego quedarnos con los cercanos, en la práctica esto es complicado. Cuando se quiere conocer la distancia entre cualesquiera dos hoteles de entre, supongamos,  $N$  opciones, es necesario almacenar  $(N^2 - N)/2$  números (una de las porciones triangulares de la matriz de distancias, quitando la diagonal). Para  $N$  incluso moderadamente grande, la cantidad de información a guardar se vuelve prohibitiva. Para empeorar las cosas, los criterios de similitud que desarrollaremos más adelante no son simétricos, por lo que tendríamos que almacenar toda la matriz (menos la diagonal), lo que resultaría en  $N^2 - N$  números.

El problema con lo anterior es que además la gran mayoría de los cálculos se desperdiciarían, ya que en nuestro contexto no tiene mucha utilidad guardar la similitud entre un hotel en Veracruz y otro en Las Vegas. Una manera de resolver este problema es ignorar hoteles que seguro no recomendaremos. Dado que queremos que el sistema regrese hoteles similares *y* cercanos, es razonable ignorar los que estén más lejos que una cantidad fija suficientemente grande en cualquier destino.

Hay diversas fórmulas para calcular la distancia geodésica entre dos puntos, unas más sofisticadas que otras. Sin embargo, como en este caso estamos interesados en hoteles cercanos, basta con la versión que supone que la Tierra es esférica:

**Definición 2.2** (Distancia geodésica). *Definimos la distancia geodésica (en kilómetros) entre dos puntos  $x = (x_{long}, x_{lat})$  y  $y = (y_{long}, y_{lat})$  por la distancia mínima que hay que recorrer por la superficie terrestre para llegar de uno a otro, suponiendo que la tierra es esférica. Si definimos  $R = \text{radio promedio de la Tierra} \approx 6371\text{km}$ ,  $\Delta_{long} = y_{long} - x_{long}$  y  $\Delta_{lat} = y_{lat} - x_{lat}$ , entonces*

$$d_{geo}(x, y) = 2R \arcsin \left\{ \sin^2(\Delta_{lat}/2) + \cos(x_{lat}) \cos(y_{lat}) \sin^2(\Delta_{long}/2) \right\}$$

Ya con una noción de distancia geográfica establecida, podemos proceder con el filtro simple propuesto.

**R 2** (Criterio de distancia estática). *Sea  $d_{geo}(i, j)$  la distancia entre los hoteles  $i$  y  $j$ , en kilómetros, y sea  $r_{outer} > 0$ . Entonces excluimos de las recomendaciones a todos los hoteles más lejanos de  $i$  que  $r_{outer}$ . Los hoteles que pasan a la siguiente etapa están contenidos en el conjunto*

$$\mathcal{H}_i^2 = \{j \in \mathcal{H}_i^1 : d_{geo}(i, j) \leq r_{outer}\}$$

*El círculo centrado en  $i$  de radio  $r_{outer}$  define lo que denominamos **cerca exterior**, fuera de la cuál ignoramos a todos los hoteles; denominamos **candidatos iniciales** a los hoteles contenidos en  $\mathcal{H}_i^2$ .*

Buscamos que  $\mathcal{H}_i^2$  sea mucho más pequeño que  $\mathcal{H}_i^1$ , ya que como la cantidad de los números a almacenar crece cuadráticamente, entonces el número de comparaciones que deberemos hacer se reducirá masivamente y por lo tanto problema será factible computacionalmente.

Es importante hacer una buena elección de  $r_{outer}$  para mantener el cálculo manejable pero sin ignorar hoteles interesantes. Un criterio que en la práctica funcionó bien es  $r_{outer} = 30$  km. En las zonas muy densas este tope nunca se alcanzará, como veremos más adelante, pero en las zonas ralas (y por ende con menos tráfico), difícilmente representará un viaje de más de media hora entre las opciones.

## 2.4. Similitud de hoteles

El filtro de cercanía y el límite de precio son esenciales para el algoritmo de recomendación. Sin embargo, no capturan nada sobre la naturaleza misma del hotel. Para ello utilizaremos información sobre los servicios disponibles en cada hotel, puesto que incluyen información más detallada sobre la categoría

y el perfil que las estrellas por sí solas. Buscaremos desarrollar una medida de similitud entre hoteles que sea más informativa y utilizarla para ordenar las recomendaciones que estén a una distancia y precio razonables.

Lo primero que tenemos que hacer al diseñar la similitud entre dos hoteles es pensar qué entendemos por “hoteles similares”. Una idea es utilizar los servicios tal cuál en un vector binario  $s_j$  para cada hotel  $j$ , y por ejemplo utilizar como medida de similitud la cantidad de servicios en que coinciden dos hoteles  $i$  y  $j$ :

$$sim(i, j) = \sum_k 1 - |s_{ik} - s_{jk}|$$

Después de algunas pruebas con distancias de este estilo nos dimos cuenta de tres problemas fundamentales que tiene este enfoque, independientemente de la medida particular que utilicemos:

1. Es una técnica muy ruidosa y no permite que servicios (casi) sustitutos contribuyan a la similitud entre los hoteles.
2. No se toma en cuenta cuál de los hoteles tiene más servicios.
3. No necesariamente se captura la filosofía de los hoteles.

Elaboremos más en estos puntos. El primero tiene que ver con el hecho de que los hoteles son más que la suma de sus servicios: son experiencias completas. Es decir, si el hotel A tiene futbol, snorkel y tenis, probablemente se parezca más al hotel B que tiene basquetbol, kayak y squash, que al hotel C, que sólo tiene futbol, puesto que aunque A y B no tienen ningún deporte en común, tienen una selección que será suficientemente buena en la mayoría de los casos. Sin embargo, cualquier similitud basada en los vectores  $s_j$  diría que A se parece más a C que a B. La solución a este problema es aglomerar los servicios en categorías dentro de las cuales los servicios sean aproximadamente equivalentes y comparar la cantidad de cada categoría.

El segundo punto es sumamente importante porque hay un factor fundamentalmente asimétrico en la forma en que las personas comparan dos hoteles. Esto se debe a que un cliente que ve un hotel con pocos servicios probablemente no le molestará que una recomendación tenga más servicios, siempre y cuando el precio y el perfil del hotel sean similares. De hecho, probablemente preferirá una mayor cantidad de servicios si se respeta el precio.

El tercer punto surge de que este tipo de medidas no capturan explícitamente la similitud entre los perfiles de dos hoteles. Si bien, mientras más

servicios coincidan, probablemente haya una mayor coincidencia, una similitud más o menos chica no necesariamente implica que dos hoteles sean fundamentalmente distintos. Consideremos dos hoteles de playa, pero uno de tres y otro de cinco estrellas. Es razonable pensar que sus perfiles no son tan lejanos entre sí como lo serían con un hotel citadino, incluso a pesar de que sus servicios probablemente diferirán más entre sí que entre el de cinco estrellas y el citadino.

Una última observación es que aunque es más fácil pensar en similitudes entre los hoteles, matemáticamente es más práctico trabajar con distancias. La razón es que ya hay una noción formal de lo que es una distancia, lo que nos permitirá utilizar matemáticas bien difundidas en lugar de inventar nociones propias:

**Definición 2.3** (Distancia). *Sea  $A$  un conjunto no vacío. La función  $d : A \rightarrow \mathbb{R}^+ \cup \{0\}$  es una distancia si y sólo si tiene las siguientes propiedades para cualesquiera  $a, b, c \in A$ :*

$$d(a, b) \geq 0 \text{ (positividad)} \quad (2.1)$$

$$d(a, b) = 0 \text{ si y sólo si } a = b \text{ (distancia mínima)} \quad (2.2)$$

$$d(a, b) = d(b, a) \text{ (simetría)} \quad (2.3)$$

$$d(a, b) \leq d(a, c) + d(c, b) \text{ (desigualdad del triángulo)} \quad (2.4)$$

Una ventaja inmediata aparte de facilitar la comprensión es que si decidimos violar algunas de las propiedades de las distancias sabremos exactamente qué estamos haciendo. Aparte de eso, este enfoque no tiene ningún inconveniente si se construye una distancia acotada, pues se puede invertir al final para generar una medida de similitud.

A continuación describiremos con más detalle cómo tomamos en cuenta estas observaciones y cómo las integramos para construir nuestra medida de similitud

## Categorización de los servicios

Dado que el enfoque servicio a servicio tiene problemas independientemente de la distancia que elijamos, será el primero que abordaremos. La manera en la que decidimos solucionarlo fue agrupar los servicios en distintas categorías. Dentro de cada una de ellas suponemos que los servicios son sustitutos perfectos, esto es, son perfectamente intercambiables. Si bien este

supuesto podría parecer algo aventurado, tuvo muy buenos resultados en la práctica y, en la mayoría de los casos, sí representa a los hoteles aproximadamente como lo haría una persona. Comenzaremos por describir la información disponible y luego la forma en que agrupamos en categorías.

**Definición 2.4** (Conjunto de servicios). *Definimos  $\mathcal{S}$  como el conjunto de todos los servicios y  $n_s = |\mathcal{S}|$ .*

Es conveniente tener un solo objeto que contenga la información de todos los hoteles y sus respectivos servicios. Elegimos una matriz para facilitar la notación.

**Definición 2.5** (Matriz de servicios). *Definimos la matriz de servicios  $S$  de  $n_s \times N$  como sigue:*

$$s_{ij} = \begin{cases} 1 & \text{si el hotel } j \text{ tiene el servicio } i \\ 0 & \text{en otro caso} \end{cases}$$

Las columnas de  $S$  corresponden a los vectores de características de los hoteles:

$$S = [s_1 s_2 \dots s_N],$$

es decir,  $s_j$  corresponde al vector de características del hotel  $j$ . Por practicidad, denotamos los renglones de  $S$  por  $z_i^T = [s_{i1}, s_{i2}, \dots, s_{i3}]$ .

Para agrupar los servicios tomamos los renglones de  $S$  como individuos y los aglomeramos utilizando un aglomerado jerárquico con el criterio de Ward, que está descrito con detalle en [5] y en [8]. Tenemos entonces  $n_s$  “observaciones” cada una con  $N$  “características”. Como en este caso sí importa tener grupos interpretables, asistimos al método manualmente. El resultado es que el algoritmo dio los grupos básicos, pero utilizamos conocimiento humano para arreglar las inconsistencias. Durante el proceso, elegimos un número de categorías y le dimos un nombre apropiado a cada una. Con lo que acabamos entonces es con un mapeo de cada servicio a su categoría correspondiente. Formalizamos estos conceptos en la siguiente definición:

**Definición 2.6** (Categorías de servicios). *Denotamos por  $\mathcal{C}$  al conjunto de todas las categorías de servicios, de cardinalidad  $n_c$ . Definimos la categoría a la que pertenece cada servicio vía la función  $\kappa : \mathcal{S} \rightarrow \mathcal{C}$ , es decir si el servicio  $s$  pertenece a la categoría  $c$ , entonces  $\kappa(s) = c$ .*

Ya teniendo el mapeo de los servicios crudos a las categorías, es conveniente generar una nueva versión de  $S$ . Dado que dentro de cada categoría supondremos que los servicios son intercambiables, basta comparar el número de servicios en cada categoría que tiene cada hotel, de modo que resulta natural la siguiente definición.

**Definición 2.7** (Matriz de categorías). *Definimos la matriz de categorías  $C$  de  $n_c \times N$  como sigue:*

$$c_{ij} = \sum_{\kappa(z)=i} s_{zj}$$

*Es decir,  $c_{ij}$  resume el número de servicios que tiene el hotel  $j$  en la categoría  $i$ . Similarmente al caso de  $S$ , denotamos las columnas de  $C$  por  $c_j$ , de modo que*

$$C = [c_1 c_2 \dots c_N]$$

De este modo, podemos utilizar las columnas de  $C$  para comparar los diversos hoteles. En este caso, por ejemplo  $\|c_i - c_j\|_1 = \sum_k |c_{ki} - c_{kj}|$  resulta en una distancia entre los hoteles  $i$  y  $j$  que permite intercambiar los servicios en cada categoría. En las siguientes secciones describiremos qué distancia decidimos utilizar.

## Similitud de servicios

Dado que el problema principal ha sido tratado, podemos proceder a definir una distancia entre los hoteles vía sus vectores característicos  $c_j$ . Una idea simple sería utilizar algo como la norma 1,  $\|\cdot\|_1$ . Sin embargo, cualquier distancia de este tipo que elijamos será simétrica, y ya establecimos la necesidad de una medida de similitud que sea asimétrica. Recordando el segundo problema fundamental de la lista veremos que es importante tomar en cuenta la diferencia entre los dos escenarios posibles, a saber, que el hotel buscado tenga más servicios que el recomendado (y por lo tanto la recomendación no es buena), o por el contrario, que el hotel recomendado tenga más servicios (en cuyo caso la recomendación es de mejor calidad). En el gráfico (2.1) ilustramos lo que sucede.

Lo que buscaremos entonces es penalizar los servicios que le faltan a un hotel a la hora de buscar candidatos. Para ello definimos la siguiente función:

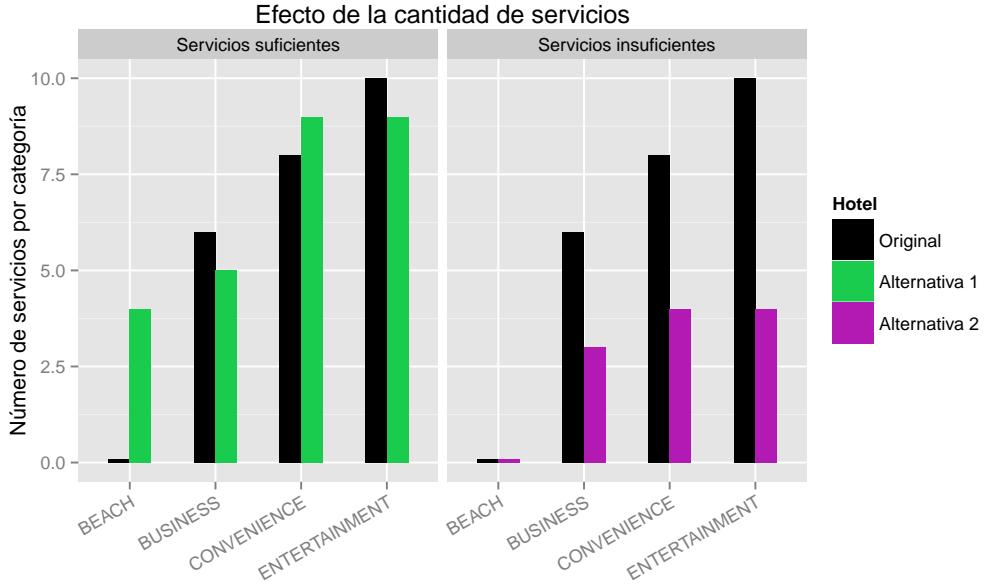


Figura 2.1: Ejemplos de similitud de servicios para 4 categorías distintas. El hotel verde se parece al negro porque no le faltan muchos servicios que el negro sí tiene. En contraste, al hotel morado le faltan muchos servicios.

**Definición 2.8** (Pseudodistancia *hinge*). Sean  $x, y \in \mathbb{R}^n$ . Definimos la pseudodistancia hinge de  $x$  a  $y$  como sigue:

$$d_h(x, y) = \sum_{k=1}^n \max\{x_k - y_k, 0\}$$

Esta función de pérdida no cumple la propiedad (2.3) porque no es simétrica, como es evidente en la figura (2.2). Por lo tanto, se trata de una *pseudodistancia*. Esta propiedad de  $d_h(\cdot, \cdot)$  es algo positivo en el contexto de nuestro problema porque refleja justamente la asimetría de la relación entre los servicios del hotel buscado y los recomendados. Aplicando esta pseudodistancia a los vectores característicos de dos hoteles  $c_i$  y  $c_j$  obtenemos:

$$d_h(c_i, c_j) = \sum_{k=1}^{n_c} \max\{c_{ki} - c_{kj}, 0\}$$

La lógica detrás de esta cantidad es la siguiente: la resta  $c_{ki} - c_{kj}$  representa la diferencia en número de servicios que el hotel  $i$  tiene sobre el  $j$  en la

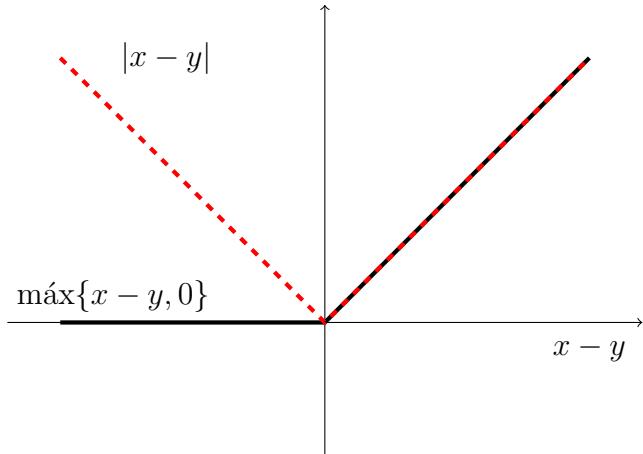


Figura 2.2: Ejemplo univariado. Distancia entre  $x$  y  $y$  usando valor absoluto (- - -) y pérdida *hinge* (—). La pérdida *hinge* no penaliza el caso en que  $y > x$ .

categoría  $k$ . Dado nuestro supuesto de que los servicios son perfectamente intercambiables, tenemos que  $c_{ki} \geq c_{kj}$  si y sólo si el hotel  $i$  es “mejor” que el  $j$  en la categoría  $k$ , y por lo tanto  $j$  incurrirá en una penalización de ese número por su falta de servicios. En el caso en que  $i$  sea peor que  $j$ , es decir, cuando  $c_{ki} \leq c_{kj}$ , no se penaliza a  $j$  y el máximo hace que se sume 0 a la distancia. En conclusión,  $d_h(c_i, c_j) = \text{cantidad de servicios que } j \text{ no tiene pero } i \text{ sí, tomando en cuenta los servicios en cada categoría como sustitutos perfectos.}$

Para etapas posteriores, cabe mencionar que dado que el número de servicios y de categorías es finito, hay una cota máxima a la distancia *hinge* entre dos hoteles, digamos  $M_h$ , que además coincide con el número total de servicios  $n_s$  (el caso se da cuando un hotel no tiene servicios y el otro tiene todos los servicios disponibles). Si bien  $d_h(\cdot, \cdot)$  tiene una interpretación muy conveniente, cambiarla de escala es más útil para el modelo y hace que sea casi inmediato ver si una distancia es grande o chica, independientemente de la cantidad de servicios en el sistema. Para ello simplemente la dividimos entre  $M_h$ , de modo que queda entre 0 y 1, lo que también permite definir una similitud asociada:

**Definición 2.9** (Pseudodistancia *hinge* normalizada y similitud de servicios). Sean  $x, y \in \mathbb{R}^n$ . Definimos la pseudodistancia *hinge* normalizada de  $x$

a y como sigue:

$$d_{hn}(x, y) = \frac{1}{n_s} d_h(x, y)$$

Definimos la similitud de servicios (o basada en hinge) entre  $x$  e  $y$  como sigue:

$$sim_h(x, y) = 1 - d_{hn}(x, y)$$

## Similitud de perfil

Si bien la pseudodistancia *hinge* captura muy bien la falta de servicios de los hoteles candidatos, no es capaz de diferenciar adecuadamente los perfiles de los hoteles. En la gráfica (2.3) ilustramos este hecho con dos ejemplos.

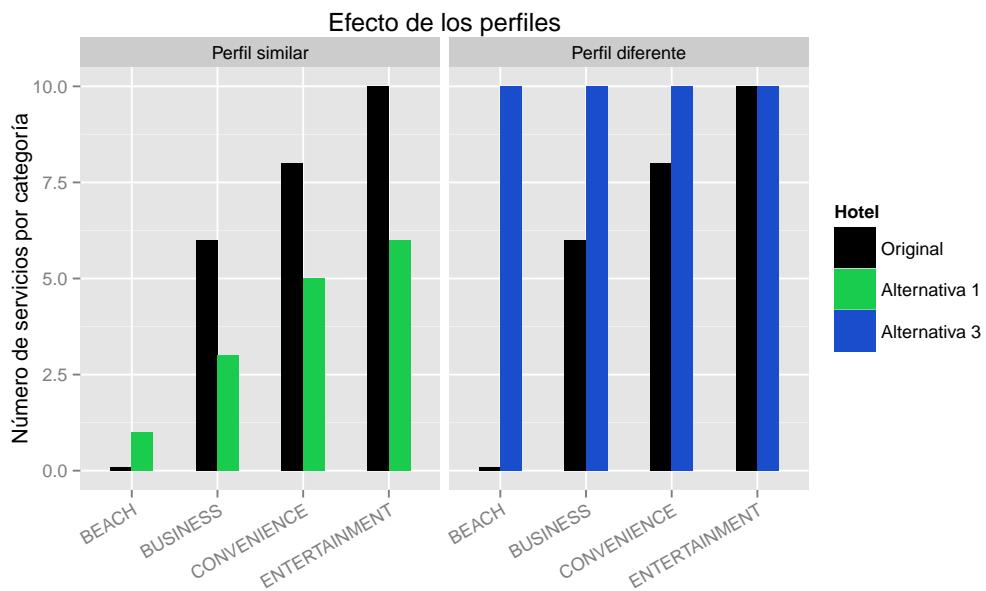


Figura 2.3: Ejemplos de similitud de perfil para 4 categorías distintas. El hotel verde se parece al negro porque aunque le faltan servicios, tiene aproximadamente la misma proporción de sus servicios en cada categoría. El hotel azul tiene muchos servicios, pero tiene un enfoque distinto (en este caso es un hotel de súper lujo muy enfocado a playa, mientras que el negro no tiene playa).

Una manera de modelar el perfil de un hotel es interpretando el número de servicios en cada categoría no como un número absoluto, sino como una *proporción* de los servicios disponibles. Es decir, tener  $m$  servicios en la categoría  $k$  es mucho o poco dependiendo del número total de servicios que tiene el hotel. Como en este caso cualquier diferencia de perfil afecta, necesitamos una distancia simétrica, que definimos a continuación.

**Definición 2.10** (Divergencia y similitud de perfil). *Sean  $x, y \in \mathbb{R}$ . Definimos la divergencia entre  $x$  e  $y$  como sigue:*

$$d_{div}(x, y) = \frac{1}{2} \sum_{k=1}^n \left| \frac{x_i}{\sum_j x_j} - \frac{y_i}{\sum_j y_j} \right|,$$

es decir, la distancia absoluta entre los vectores normalizados por su suma. Definimos también la similitud de perfil (o basada en la divergencia) como

$$sim_{div}(x, y) = 1 - d_{div}(x, y)$$

Como  $d_{div}(\cdot, \cdot)$  siempre está entre 0 y 1, La ventaja de esta distancia (se deja como ejercicio al lector probar que en efecto cumple todas las propiedades) es que primero quita la escala y luego compara los vectores en un espacio equitativo. Regresando a la gráfica (2.3), notamos que bajo este criterio, el hotel verde es prácticamente idéntico al negro, incluso a pesar de que es un hotel de mucho menor categoría, debido a que tienen una distribución similar de servicios en cada una de las categorías. Por otra parte, aunque el azul tiene más servicios en cada categoría, tiene una distribución más uniforme que el negro y, por lo tanto, su distancia basada en divergencia será grande.

Finalmente, dado que los vectores normalizados suman 1 y los estamos restando, es fácil ver que por el factor de  $1/2$  que incluimos en la definición, la divergencia nunca puede pasar de 1 y por lo tanto la similitud que definimos en efecto tiene sentido.

## Similitud completa

En las secciones anteriores resolvimos parcialmente los tres puntos críticos. Por un lado, al mudar el análisis a la escala de categorías resolvimos el problema del ruido y, por otro, cada uno de los criterios de similitud de servicios y de perfil resuelve uno de los otros dos problemas. Sin embargo, como explicamos en cada sección, ambas tienen fallas. Lo que buscamos entonces

es integrar las dos medidas en un criterio que sí tome en cuenta todo. La dificultad radica en cómo hacerlo sin perder información y dándole un peso apropiado a cada uno.

De este punto en adelante trabajaremos nuevamente con similitudes, porque es más cómodo hablar de similitudes y porque además son equivalentes a sus distancias asociadas. Dado que tanto la similitud de servicios como la de perfil están en escalas equiparables, una manera razonable de combinarlas sería promediarlas. Sin embargo, podrían tener diferentes niveles de variabilidad, por lo que es prudente permitir la posibilidad de utilizar una ponderación distinta de  $1/2$ .

**Definición 2.11** (Similitud entre hoteles). *Sea  $\alpha \in (0, 1)$  y sean  $i, j$  hoteles en  $\mathcal{H}$ . Definimos la similitud con parámetro  $\alpha$  entre  $i$  y  $j$  como sigue:*

$$sim_\alpha(c_i, c_j) = \alpha sim_h(c_i, c_j) + (1 - \alpha) sim_{div}(c_i, c_j)$$

Cabe mencionar que la similitud entre hoteles no es simétrica porque está basada en parte en la similitud de servicios. Es decir, no necesariamente se cumple que  $sim_\alpha(c_i, c_j) = sim_\alpha(c_j, c_i)$ .

Ya habiendo establecido la forma de combinar las dos medidas de similitud, el siguiente paso es elegir un valor apropiado de  $\alpha$ , que de preferencia sea óptima con respecto a algún criterio. Para ello, hay que notar que aunque ambas similitudes están entre 0 y 1, podrían tener niveles de variación y medias distintos, por lo que utilizar  $\alpha = 1/2$  podría no ser una buena idea. Ilustramos este hecho en la figura (2.4). En el ejemplo,  $\alpha = 1/2$  probablemente sería similar a avanzar en la flecha pequeña de la derecha, pero es claro que avanzando en la dirección de la de en medio estaríamos tratando ambas similitudes de manera más equitativa. Dado que lo que queremos es identificar hoteles similares, una buena manera de decidir el valor de  $\alpha$  es utilizando la que maximice la varianza explicada por la similitud completa. Proponemos utilizar un Análisis de Componentes principales para calcular la  $\alpha$  óptima. Al lector interesado en esta técnica lo remitimos a [2], en donde encontrará además diversas variantes del mismo.

Como queremos calcular el valor óptimo de  $\alpha$  para las recomendaciones que haremos, tenemos que hacer el análisis con todas (o una muestra suficiente de) las recomendaciones de todos los hoteles. La forma de hacerlo es como sigue:

1. Establecer los parámetros de los filtros estáticos de precio y distancia.

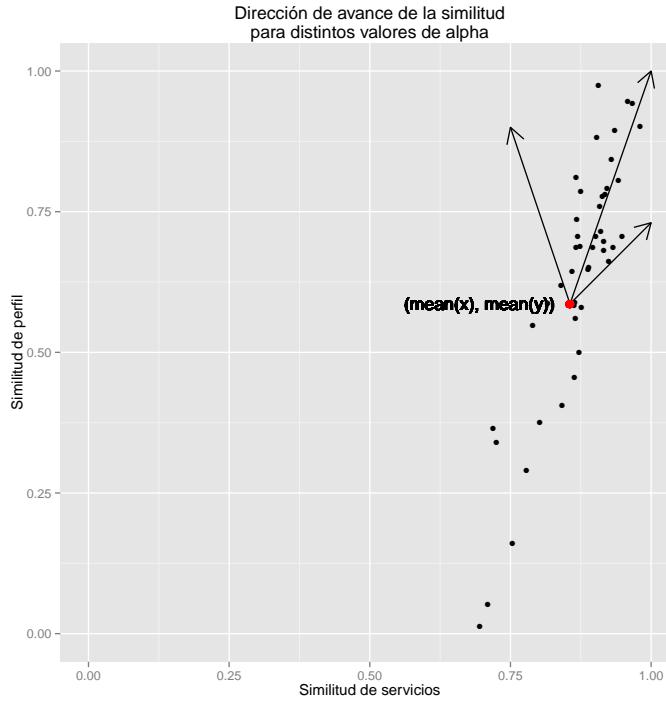


Figura 2.4: La elección de  $\alpha$  determina la dirección en la que avanza la similitud total al variar la de servicios y la de perfil.

2. Para cada hotel  $i$ , generamos  $\mathcal{H}_i^2$ .
3. Para todo  $j \in \mathcal{H}_i^2$ , calculamos

$$\begin{aligned} h_{ij} &:= \text{sim}_h(c_i, c_j) \\ d_{ij} &:= \text{sim}_{\text{div}}(c_i, c_j) \end{aligned}$$

4. Insertamos cada pareja  $(h_{ij}, d_{ij})$  como un renglón en una matriz  $A$ , con dos columnas y número de renglones igual a  $\sum_i |\mathcal{H}_i^2|$ . Es decir, los renglones de  $A$  pueden ser representados como en la figura (2.4).
5. Corremos Componentes Principales sobre  $A$  y obtenemos el vector propio asociado a la primera componente principal, digamos

$$\vec{a} = (a_h, a_d)$$

6. El vector  $\vec{a}$  define la dirección deseada. Para que sus componentes sumen 1 y por lo tanto definan valores aceptables para  $\alpha$  y  $1-\alpha$ , dividimos sus componentes entre la suma, de modo que:

$$\begin{aligned}\alpha^* &:= \frac{a_h}{a_h + a_d} \\ 1 - \alpha^* &:= \frac{a_d}{a_h + a_d}\end{aligned}$$

De este punto en adelante únicamente utilizaremos la similitud completa utilizando  $\alpha^*$ . La siguiente definición nos ayudará a facilitar la exposición.

**Definición 2.12** (Score). *La similitud completa que resulta de utilizar  $\alpha^*$  se denomina score, es decir:*

$$score(i, j) := sim_{\alpha^*}(c_i, c_j)$$

El *score* condensa todas las propiedades que buscábamos en una medida de similitud y que no tenga los tres problemas expuestos al principio de esta sección. Claramente no presentará la volatilidad de las medidas basadas en servicios crudos porque está basada en categorías. Sin embargo, este mismo hecho genera potencial de enmascaramiento si las categorías no están bien elegidas. Esto se debe a que si son heterogéneas y no consisten de servicios que sean sustitutos más o menos cercanos, puede haber falsos positivos y falsos negativos al calcular cualquier medida basada en ellos. En cuanto al segundo punto, suponiendo que las categorías en efecto son apropiadas, la parte de *hinge* del *score* garantizará hasta cierto punto que no se recomiende hoteles con demasiado pocos de los servicios de interés. Finalmente, el hecho de que haya una colaboración de la similitud basada en la divergencia evita que se recomiende hoteles con un perfil muy distinto al original.

Suponiendo que el *score* es una medida satisfactoria de similitud, podemos proseguir a los siguientes componentes del sistema de recomendación, que tomarán en cuenta aspectos físicos y temporales que son imposibles de capturar utilizando únicamente información de los servicios de los hoteles.

## 2.5. Distancia: filtro dinámico

El concepto de “hotel similar” que buscamos en este caso tiene que ver no sólo con que los hoteles sean en sí mismos similares, sino que además estén

cerca. La dificultad es que la noción de lo que significa la palabra “cerca” varía de lugar a lugar. Por ejemplo, en una zona con una alta densidad de hoteles, como Reforma Centro en la Ciudad de México, un hotel es considerado cercano si está a, digamos, no más de 1 ó 2 kilómetros. Una opción en el Aeropuerto Benito Juárez no es aceptable y difícilmente alguien diría que dicha opción está cerca de Reforma Centro (aunque en realidad está a 15 km). Sin embargo, en una zona despoblada como la Riviera Maya, en la que todo el transporte es por carretera, 15 kilómetros son 10 minutos en coche, y de hecho hay muchos hoteles que no tienen vecinos a menos de varios kilómetros de distancia. La moraleja es que la distancia se percibe distinto según la zona, por lo que debemos utilizar algún criterio adaptativo.

Una primera idea para incorporar la distancia en el modelo fue meterla de alguna manera en el *score*, por ejemplo, como un tercer término en el promedio. Sin embargo, encontramos un argumento que sugiere que esto podría no ser la mejor idea: Supongamos que buscamos recomendaciones para el hotel A y tenemos dos candidatos, B y C. Dada la densidad de la zona, ambos están “cerca” de A, pero resulta que B está a la mitad de distancia de A que C. Además, supongamos que C se parece un poco más a A que B. La pregunta es entonces: ¿deberíamos recomendar a B sobre C aunque se parece menos y dado que ambos están cerca? Creemos que la respuesta a este ejercicio es claramente que no, que si ambos están cerca, es preferible el hotel más similar. Esta manera de pensar nos llevó a la conclusión de que incorporar la distancia de manera suave (como por ejemplo dándole una ponderación en el *score*) tal vez no sea la mejor alternativa. Por ello decidimos mejor incorporarla en la forma de un filtro dinámico. Para ello hay que seguir dos pasos:

1. Definir qué entendemos por hoteles cercanos.
2. Pasar a la siguiente ronda del modelo los hoteles que cumplen dicho criterio, ordenados por *score*, es decir, por qué tan parecidos son al hotel en cuestión.

El paso crítico para este método es la definición de cercanía. Una primera idea es, por ejemplo, elegir los 20 ó 30 hoteles más cercanos al hotel en cuestión. Sin embargo, este enfoque ingenuo tiene un problema grave: si un hotel es único en su categoría en la zona (por ejemplo es de 5 estrellas en un pueblo pequeño), entonces posiblemente los 20 hoteles más cercanos a él no se parezcan en lo absoluto. En estos casos debería permitirse una mayor

flexibilidad de búsqueda en favor de obtener mejores recomendaciones. En la figura (2.5) ilustramos lo que sucede y a continuación describimos el proceso formalmente.

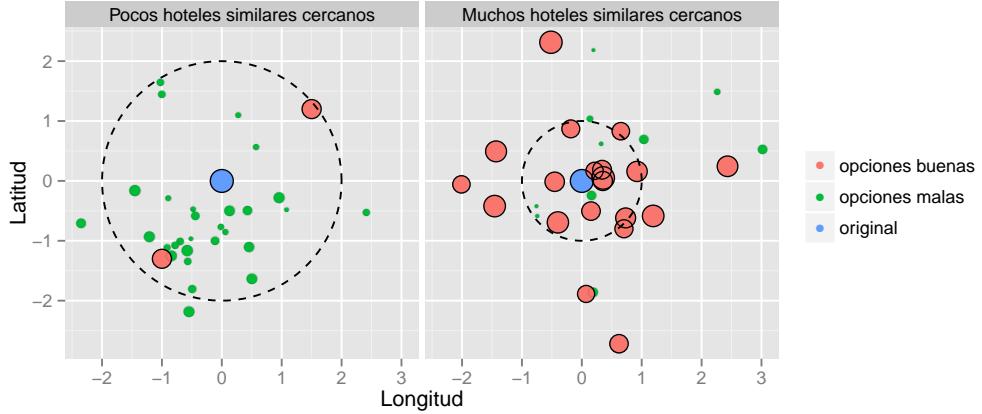


Figura 2.5: El filtro debe ser más laxo cuando hay pocos hoteles similares cercanos (izquierda) y más estricto cuando hay más (derecha).

**R 3** (Distancia: filtro dinámico). *La función de acumulación de score al alejarse del hotel  $i$ ,  $F_i : \mathbb{R} \rightarrow \mathbb{R}$ , está definida para cada  $d \in \mathbb{R}$  como sigue:*

$$F_i(d) = \sum_{j|d_{geo}(i,j) < d} \text{score}(i, j)$$

*Para una cantidad predefinida a acumular, digamos  $w_{inner} > 0$ , definimos la cerca interior como el círculo centrado en el hotel  $i$  y con radio dado por*

$$r_{inner} = \arg \min_d \{F_i(d) \geq w_{inner}\},$$

*es decir, la mínima distancia que garantice que se acumula al menos  $w_{inner}$  de score dentro de la cerca interior. Los hoteles que pasan a la siguiente ronda son entonces los que estén en la cerca interior:*

$$\mathcal{H}_i^3 = \{j \in \mathcal{H}_i^2 : d_{geo}(i, j) \leq r_{inner}\}$$

La justificación de la regla anterior es que cuando un hotel es único en su área, los hoteles cercanos no se parecerán a él, lo que provocará que  $F_i$  crezca lentamente y, por consiguiente, que  $r_{inner}$  sea grande. Por el contrario, si hay muchos hoteles parecidos a  $i$  cerca, entonces  $F_i$  crecerá rápidamente y por lo tanto  $r_{inner}$  será más pequeño.

## 2.6. Precios dinámicos

Un punto que hasta el momento no hemos tratado son las posibles promociones que los hoteles puedan ofrecer. Si bien el objetivo principal de este sistema es recomendar hoteles similares y no necesariamente descuentos, es deseable que éstos tuvieran algún impacto. Para ello notemos que el filtro estático de precio toma en cuenta el promedio de los precios en el largo plazo. Eso quiere decir que si un hotel baja sus precios temporalmente, podría ser un candidato para otros hoteles más baratos, pero tanto la baja de los precios como su impacto se ven diluidos en el promedio. Una forma de suavizar este problema es filtrar con los precios en la fecha de la búsqueda en lugar de los promedios. Es decir que, si bien debemos calcular  $\alpha$  y  $r_{inner}$  con  $\mathcal{H}_i^2$  tal como lo hemos descrito hasta el momento, conviene utilizar

$$\mathcal{D}^* = \text{período para el cual se busca reservar} \quad (2.5)$$

para calcular los precios en lugar del período histórico largo  $\mathcal{D}$ .

**R 4** (Precio dinámico). *Dadas  $\alpha$  y  $r_{inner}$  obtenidas en R1, R2 y R3, denotamos las recomendaciones finales por*

$$\mathcal{R}_i = \{j \in \mathcal{H} : p_j^{D^*} \leq (1 + \gamma)p_i^{D^*}, d_{geo}(i, j) \leq r_{inner}\},$$

*y los hoteles se presentarán ordenados por score, de mayor a menor.*

Hay que tener en cuenta que este último cambio aumenta considerablemente el costo computacional de calcular las recomendaciones, ya que habrá que conseguir los precios de todos los hoteles cercanos a  $i$  en las fechas apropiadas cada vez que un usuario entre a la página del hotel (sería inviable almacenar todas las recomendaciones para todas las fechas posibles). Comárese esto con utilizar precios estáticos de largo plazo y almacenar las recomendaciones de  $i$  una sola vez, dado que serán fijas para todos los usuarios.

Si se considera que la carga computacional es demasiado pesada para la infraestructura en la que se correrá el sistema de recomendación, se puede omitir los precios dinámicos, poniendo  $\mathcal{R} := \mathcal{H}_i^3$  y mostrando los hoteles ordenados por *score* de la misma manera que en R4.

## 2.7. Consideraciones prácticas

Una última consideración que hay que tener son los casos extremos. Un caso es que la zona del hotel buscado tenga muy pocas opciones dentro de la cerca exterior y el otro es cuando el hotel es muy barato para la zona, por lo que el filtro de precio le impide ver suficientes de los demás hoteles. En ambos, un asesor deberá utilizar su *ingenio* para salir de paso. Lo que es indispensable es que *siempre* recomiende algo, aunque sea lo menos peor cuando no tenga alternativa.

Para aliviar el primer problema el procedimiento es simple: como no queremos que se queden vacías las recomendaciones, ignoramos todo lo anterior y regresamos un número pequeño predefinido de hoteles cercanos, ordenados por distancia. Decidimos ignorar el *score* en estos casos porque, dado que hay muy pocas opciones, creemos que hay que darle prioridad al lugar en particular que buscó el cliente. Si, por otro lado, sí hay suficientes hoteles dentro de la cerca exterior, pero el filtro de precio es demasiado estricto, entonces ignoramos por completo este filtro.

**R 5** (Casos extremos). *Sea  $rec_{min} > 0$  el mínimo número de recomendaciones que se desea.*

(I) *Si  $|\mathcal{H}_i^2| < rec_{min}$ , entonces redefinimos*

$$\mathcal{H}_i^1 = \mathcal{H}$$

(II) *Si aún después del paso anterior  $|\mathcal{H}_i^2| < rec_{min}$ , entonces ignoramos todas las reglas y recomendamos los hoteles en*

$$\mathcal{R}_i^{ex} = \{j \in \mathcal{H} : j \text{ es uno de los } rec_{min} \text{ hoteles más cercanos a } i\}$$

*Además, ignoramos el score y los presentamos ordenados por distancia al hotel  $i$ .*

(III) *Las dos reglas anteriores aplican de manera análoga para  $\mathcal{H}_i^{1*}$  y  $\mathcal{H}_i^{2*}$ , los equivalentes a  $\mathcal{H}_i^1$  y  $\mathcal{H}_i^2$  que utilizan el precio dinámico.*

## 2.8. Juntando todo

El procedimiento descrito anteriormente es largo y puede ser difícil de asimilar en su conjunto, de modo que presentamos aquí una versión resumida:

1. Sean  $i$  un hotel,  $D$  un periodo largo y  $D^*$  el periodo en el que se desea reservar.
2. Calculamos  $\alpha$  y  $r_{inner}$  como está descrito en R1, R2 y R3.
3. Checamos los casos extremos descritos en R5. Si ninguno aplica, continuamos; si resulta el caso (II), entonces recalculamos el paso (2) y continuamos al (4); si resulta el caso (I), recomendamos con  $\mathcal{R}_i^{ex}$  por distancia y terminamos.
4. Generamos las recomendaciones finales como está descrito en R4 y las presentamos ordenadas por *score*.

## 2.9. Comentarios

Antes de terminar el capítulo, hay un punto que nos gustaría resaltar y es por qué no decidimos simplemente basar la distancia en la norma 1. La respuesta corta es que es muy exigente. Aunado a ello, no nos permite ponderar fácilmente las contribuciones del perfil y de los servicios. Es posible que haya alguna manera de alterar esta distancia para capturar la asimetría de la disponibilidad de servicios. De ninguna forma afirmamos que no haya alternativas al menos tan buenas como la nuestra, pero dado que obtuvimos resultados satisfactorios, nos quedamos con el método presentado anteriormente. Investigar nuevas medidas de distancia sería un buen tema de trabajo futuro.

# Capítulo 3

## Implementación

En este capítulo describiremos algunos detalles adicionales al modelo matemático que son necesarios para la implementación del sistema de recomendación en el mundo real. Hablaremos de qué fuentes obtendremos los datos necesarios para el modelo y de qué haremos en los casos en los que la calidad de los mismos no sea la necesaria para correr el modelo.

### 3.1. Información

En esta sección describiremos el qué, el cómo y el dónde de la información que utilizamos para el sistema de recomendación. Para ello hay que notar que necesita dos insumos para funcionar. El primero es información a nivel hotel, como son los nombres, los detalles geográficos y los precios. El segundo consiste de la información de los servicios del hotel. En los siguientes dos apartados describiremos cada uno con mayor detalle.

#### Información de hoteles

La tabla de información de hoteles debe tener por lo menos las siguientes columnas para que el modelo funcione:

- Clave del hotel.
- Latitud y longitud.
- Precio promedio por noche en alguna moneda unificada (USD) y en algún período razonablemente largo, digamos uno o dos años.

Opcionalmente se puede agregar el volumen de noches y la utilidad bruta en el mismo periodo para hacer algunos análisis, aunque el modelo estrictamente no los necesita.

Para obtener esta tabla es necesario notar que no hay un catálogo de precios, por lo que los tuvimos que estimar mediante las reservaciones que se hicieron en el periodo analizado. Es decir, el precio promedio por noche en el periodo  $D$  (en USD) de un hotel dado es:

$$\text{precio promedio} = \frac{\text{Suma de importes (USD) de las reservas en el periodo } D}{\text{Suma de noches reservadas en el periodo } D}$$

Para accesar fácilmente esta información generamos un procedimiento en SQL Server que recibe una fecha de inicio, hace  $D$  = de la fecha proporcionada hasta hoy, y finalmente regresa la tabla con la información de los hoteles. Para la primera versión decidimos utilizar un periodo de 2 años, con el fin de evitar que los hoteles poco vendidos no tuvieran ningún precio.

## Información de servicios

Las matrices  $S$  y  $C$ , que contienen la información de los servicios y las categorías respectivamente, son bastante ralas. Por esta razón es conveniente almacenarlas como relaciones con un renglón por cada elemento distinto de cero de la matriz correspondiente. Como  $S$  es binaria, la almacenaremos en una tabla con dos columnas, clave del hotel y clave del servicio, mientras que  $C$  requiere tres: clave del hotel, clave de la categoría y número de servicios en la categoría.

Dado que en la base de datos de Best Day ya está la tabla con los servicios en el formato deseado, basta filtrar los hoteles con información incompleta o de zonas que no nos interesen y ya tenemos la primera tabla. Para obtener la segunda fue necesario primero correr el aglomerado en R utilizando la primera tabla y crear a mano la relación entre los servicios y las categorías que elegimos. Posteriormente bastó pegarle a cada servicio su categoría correspondiente y sumarizar la tabla de servicios utilizando la clave del hotel y la clave de la categoría como llaves; los conteos dentro de cada grupo corresponden a los valores  $c_{ij}$ . Nuevamente, generamos un procedimiento de SQL Server para poder obtener esta información con una simple llamada a una función en lugar de una consulta completa.

A continuación describiremos un tema sobre los servicios que es importante en el contexto de la recomendación de hoteles similares.

## **Plan de alimentos**

La decisión de exactamente qué características del hotel cuentan como servicios es subjetiva hasta cierto punto. Hay algunos que no se incluyeron por razones administrativas o de alguna otra índole. Entre ellos están los planes de alimentos, es decir, la cantidad de comidas que incluyen los hoteles. El plan de alimentos puede ser muy importante en algunas regiones vacacionales en donde haya hoteles de tipo “todo incluido” mezclados con hoteles que sólo incluyen desayuno, o incluso nada. La razón es que los precios de dos hoteles de las mismas estrellas podrían diferir bastante si uno es todo incluido y el otro no. Por ello es importante incorporar esta información de algún modo en el modelo.

Si bien hay diversas formas en que se podría tomar en cuenta el plan de alimentos en el sistema de recomendación, uno que nos pareció razonable, fácil de implementar y que además funciona bien, es generar una categoría de alimentos en la matriz  $C$ . La idea es que si dos hoteles tienen el mismo plan de alimentos, tendrán más servicios en común y por lo tanto favoreceremos que se recomiendan entre sí, mientras que si difieren, se tendrán que parecer mucho para que salgan en las recomendaciones del otro. Cada hotel puede tener muchos contratos con Best Day y puede vender cuartos con y sin todo incluido, pero en general, es razonable pensar que si un hotel tiene la opción de todo incluido, es parte de la filosofía del hotel. Por ello, para cada hotel elegimos el plan de alimentos que más comidas incluidas tuviera. El resultado es que podemos agregar hasta cuatro servicios por hotel, uno si incluye desayuno, uno si comida, uno si cena y otro si incluye las bebidas. En la matriz  $C$  esto se refleja en una nueva categoría que puede tomar cualquier valor entre cero y cuatro.

## **3.2. Sistema de respaldo**

Uno de los grandes temas que surgen en cualquier modelo matemático que surja del análisis de datos es la calidad de la información. El modelo descrito en el capítulo anterior presupone no sólo que se tiene la información ya en el formato adecuado, sino que además está limpia. La norma en el mundo real es que esto no es así, por lo que será necesario agregar componentes al sistema de recomendación para que funcione incluso cuando la información no sea perfecta. A continuación enlistaremos algunos de los principales pro-

blemas que pueden aparecer en este contexto y las posibles soluciones que les podemos dar:

P1: El hotel no tiene ninguna reservación en el periodo analizado para obtener los precios.

S1: Utilizamos el modelo pero ignoramos los precios por completo.

P2: El hotel tiene mal la información de servicios.

S2: Tendremos que utilizar un sistema de recomendación alternativo, como por ejemplo el antiguo.

P3: El hotel no tiene coordenadas o las coordenadas son incorrectas.

S3: No hay nada que podamos hacer salvo conseguirlas de alguna fuente.

La primera dificultad no es tan grave porque podemos utilizar el modelo sin problemas, sólo que ignorando los precios. En realidad es lo mejor que podemos hacer y dado que de cualquier forma tiene que ser un hotel muy poco popular, no importa mucho que las recomendaciones no sean óptimas en estos casos.

Los otros dos puntos son más complicados de resolver porque implican que de plano no podemos utilizar el sistema de recomendación. Una forma de mitigar esto es utilizar el sistema de recomendación antiguo de Best Day que está descrito en el capítulo 1, es decir, tomar los hoteles en el mismo destino y de las mismas estrellas y recomendarlos en el *Sort Order* del momento. La ventaja de esto es que únicamente requiere el destino y las estrellas del hotel para funcionar, por lo que en la mayoría de los casos podremos evitar no recomendar nada.

Si también falla el sistema de recomendación antiguo no hay mucho que podamos hacer, pues o bien ni siquiera tenemos el destino o estrellas del hotel, o no hay ninguna alternativa en el mismo destino. Si detectamos una zona o algún hotel con esta clase de problemas, lo único podríamos hacer es detectar la información faltante o de mala calidad y completarla o limpiarla a mano, según lo requiera el caso.

### 3.3. Otras consideraciones

Hay algunas otros detalles técnicos que no entran naturalmente en ningún otro lado, pero que sí son importantes. En esta sección los describiremos

brevemente con el fin de aclarar por completo la forma en que se debe implementar este sistema de recomendación:

- *Software*: Como hemos comentado, el acceso a la base de datos, tanto para el análisis como para producción, se realizó a través de código SQL. En particular, utilizamos Stored Procedures de SQL Server 2008, optimizadas para evitar sobrecargar la base. En [4] hay información de las últimas versiones de este software. Toda la analítica se llevó a cabo en *R* (sitio del proyecto en [1]), desde el aglomerado jerárquico hasta la generación de las recomendaciones. En el diagrama (3.1) de la siguiente sección será más evidente qué partes del sistema se hicieron en qué lenguaje.
- *Almacenamiento de las recomendaciones*: En la práctica no es conveniente calcular los conjuntos  $\mathcal{R}_i$  al vuelo por su elevado costo computacional. Por ello los precalculamos para cada hotel y los metimos en una tabla *R* con tuplas de la forma (hotel buscado, hotel recomendado, *score*). Si se quiere utilizar los precios estáticos, se puede tomar, digamos, los 20 hoteles recomendados con mayor score para cada hotel buscado. Esto disminuye aún bastante el tamaño de *R* y, por lo tanto, el costo de consultar las recomendaciones de un hotel. Si, por otro lado, se utilizan los precios estáticos, se debe almacenar todos los hoteles en la cerca interior y filtrarlos dinámicamente con los precios de la fecha consultada.
- *Actualización de las recomendaciones*: Dado que los hoteles son entidades más o menos estáticas, no tiene mucho sentido actualizarlas demasiado seguido. Basta correr el algoritmo que genera las recomendaciones una vez al mes, cuando mucho, para ir capturando los pequeños cambios que puedan haber ocurrido.

### 3.4. Arquitectura final del sistema

En la figura (3.1) mostramos la arquitectura que el sistema debería tener idealmente. Por un lado, la base de datos analítica alimenta los Stored Procedures y el aglomerado jerárquico en *R* de los servicios en categorías. Estos procesos se encargan de generar la matriz *C* y la tabla *H* con la información de los hoteles, completa con coordenadas y precios. El algoritmo

de recomendación corre completamente en  $R$  y genera una tabla con las recomendaciones. Esta tabla se debe cargar a la base de datos operativa para entrar a producción. Idealmente, la carga se debería realizar automáticamente, pero dada la baja frecuencia de las actualizaciones, si por alguna razón no es viable hacerlo así, basta con que se pueda hacer manualmente. Una vez que la tabla  $R$  está en la base de datos operativa, la página (i.e. nuestro “vendedor” electrónico) puede aprovecharla para recomendar de manera más inteligente.

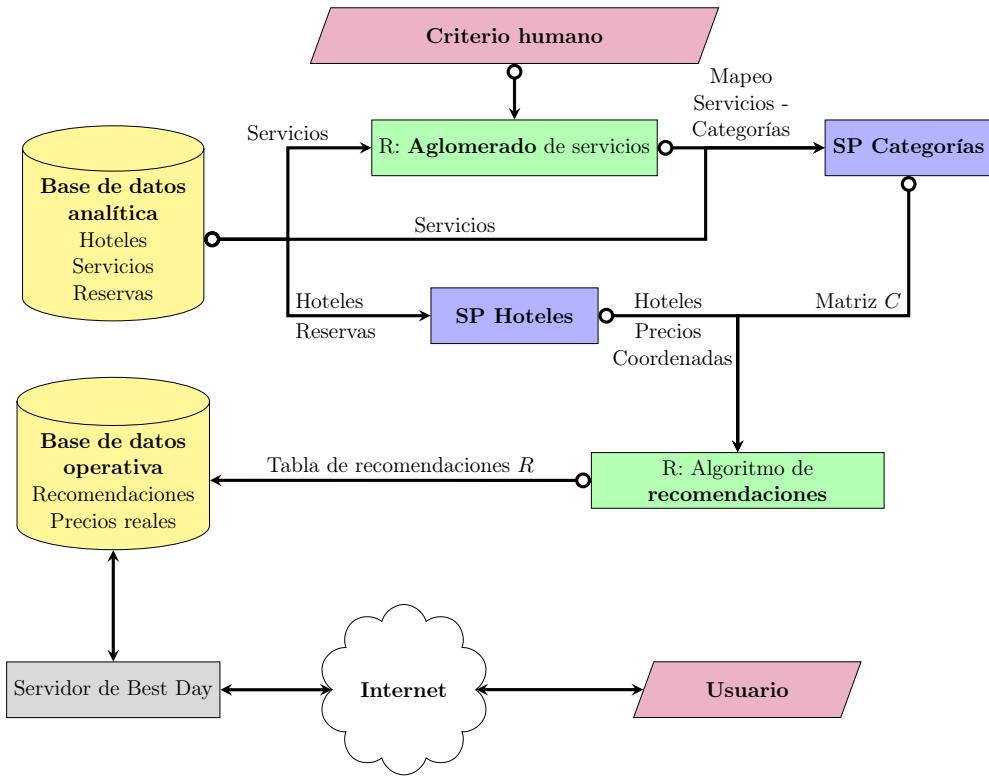


Figura 3.1: Diagrama de flujo del sistema de recomendación. Amarillo: bases de datos. Rojo: Interacción humana. Verde: Procesos en  $R$ . Azul: Procesos en SQL.

# Capítulo 4

## Desempeño

Hasta este punto hemos descrito tanto el modelo matemático detrás del sistema de recomendación como las consideraciones prácticas que se debe tener para implementarlo exitosamente. Si bien hemos argumentado todas las decisiones que tomamos en la construcción del sistema, en este capítulo analizaremos su comportamiento en la práctica desde dos enfoques: un análisis de redes y un análisis de la página web de Best Day usando Google Analytics [3]. En las siguientes dos secciones hablaremos de cada uno de ellos

### 4.1. Análisis de redes

Las redes son construcciones matemáticas muy versátiles que permiten modelar muchos tipos de fenómenos. En general, cualquier tema de interés que tenga elementos que estén conectados de alguna forma se puede modelar con una red. En las siguientes secciones daremos un brevísimo resumen de algunas de las características principales de las redes y evaluaremos el desempeño del sistema de recomendación utilizando este enfoque.

#### Teoría de redes

Antes de describir las características de las redes, debemos definir lo que son. Una red es esencialmente un conjunto  $V$  (que puede incluir cualquier tipo de entidades, individuos, etc.), junto con una colección de conexiones  $E$  entre los elementos de  $V$ . Los elementos en  $V$  usualmente se conocen como *nodos* o *vértices*, mientras que las conexiones (en  $E$ ) son llamadas *arcos*. Hay dos principales tipos de redes y sus diferencias radican en si sus arcos son *dirigidos*

o *no dirigidos*. Los arcos dirigidos se utilizan para representar relaciones simétricas, como la amistad: “Juan es amigo de Pedro si y sólo si Pedro es amigo de Juan”. Si bien no en todos los casos la reciprocidad es total, cuando en general la conexión es bilateral se recurre a las redes no dirigidas. Por otro lado, se utilizan arcos dirigidos cuando las relaciones no necesariamente son bilaterales. Por ejemplo, si se toman los cruces de calles como nodos y los arcos como las calles mismas, cada arco debe tener la dirección de la calle que representa. Cuando se da el caso de relaciones bilaterales en una red dirigida, se puede poner dos arcos, uno en cada dirección. Otras características interesantes de las redes son:

- *Variables adicionales de nodos y/o arcos:* Los nodos y los arcos pueden tener otro tipo de características. Por ejemplo, si la red representa una red de tuberías, los arcos podrían tener un flujo máximo asociado.
- *Distancia entre nodos:* La distancia entre dos nodos es la mínima cantidad de arcos que se debe atravesar para llegar de uno de los nodos al otro. Si la red es dirigida sólo está permitido ir en la dirección de los arcos.
- *Conejividad:* Una gráfica es conexa cuando se puede llegar a cualquier nodo desde cualquier otro nodo. En el caso de las gráficas dirigidas sólo se permite transitar en el sentido de los arcos.
- *Diámetro:* El diámetro de una gráfica conexa es el máximo número de pasos que se requiere para llegar de cualquier nodo a cualquier otro. Nuevamente, en las gráficas dirigidas se debe respetar la dirección de los arcos.
- *Grados:*
  - *Grado de entrada (in-degree):* En redes dirigidas, el número de arcos que le apuntan a un nodo.
  - *Grado de salida (out-degree):* En redes dirigidas, el número de arcos que salen de un nodo.
  - *Grado (degree):* En redes dirigidas, corresponde al grado de salida más el grado de entrada. En redes no dirigidas, es el número total de arcos de un nodo.

- *PageRank*: Es una medida de importancia que fue diseñada para ver la relevancia de las páginas web. A grandes rasgos, coincide con la probabilidad de que un navegador que da *click* al azar en las ligas que ve llegue a una cierta página (nodo). Utilizaremos el PageRank para medir la importancia de los hoteles dentro del sistema, pues representará la probabilidad de que un usuario llegue a un hotel dado si le da *click* a recomendaciones al azar. Al lector interesado lo referimos al artículo original, [6], y a [7] para mayor información sobre el PageRank y sobre redes en general.

## Modelado del sistema con una red

Los hoteles y sus recomendaciones se pueden modelar de manera muy natural como una red dirigida tomando los hoteles como nodos y el que un hotel recomienda a otro como los arcos. Con un análisis de redes podremos ver si realmente el sistema recomendación cumple con algunas de las propiedades que buscábamos.

Uno de los primeros problemas que le encontramos al sistema actual es que dentro de una zona y una categoría las recomendaciones eran siempre iguales, lo que además de resultar aburrido y poco útil después de poco tiempo, le da un peso excesivo a los hoteles que aparecen hasta arriba de la lista de un destino y ciertas estrellas: todos los clientes que busquen un hotel en ese grupo verán lo mismo. En el mundo de las redes, esto es equivalente a nodos con un grado de entrada muy elevado o con PageRank alto.

Por otro lado, independientemente de lo anterior, las recomendaciones eran cerradas, esto es, no podíamos salir de un cierto grupo de hoteles únicamente por medio de las recomendaciones. Esto equivale a que la red del modelo sea *disconexa* (i.e. no conexa) en lugares en donde no debería. No es problemático que, por ejemplo, la recomendaciones de hoteles en Ciudad Juárez no nos pudieran llevar eventualmente a opciones en Mérida. Lo que sí es poco deseable es que secciones geográficamente cercanas sean disconexas, como por ejemplo si empezamos con un hotel de negocios en Cancún, que no haya forma de llegar a uno en la zona hotelera. Ilustramos estos puntos en la gráfica (4.1). Para modelar las recomendaciones del sistema con una red hay que tener algunas consideraciones. Por un lado, a cada hotel le asignamos un ordenamiento de los demás hoteles dentro de un cierto radio vía el *score*. Sin embargo, utilizar esto como peso para los arcos no es muy realista, puesto que el *score* está en una escala sin ninguna interpretación. Una mejor al-

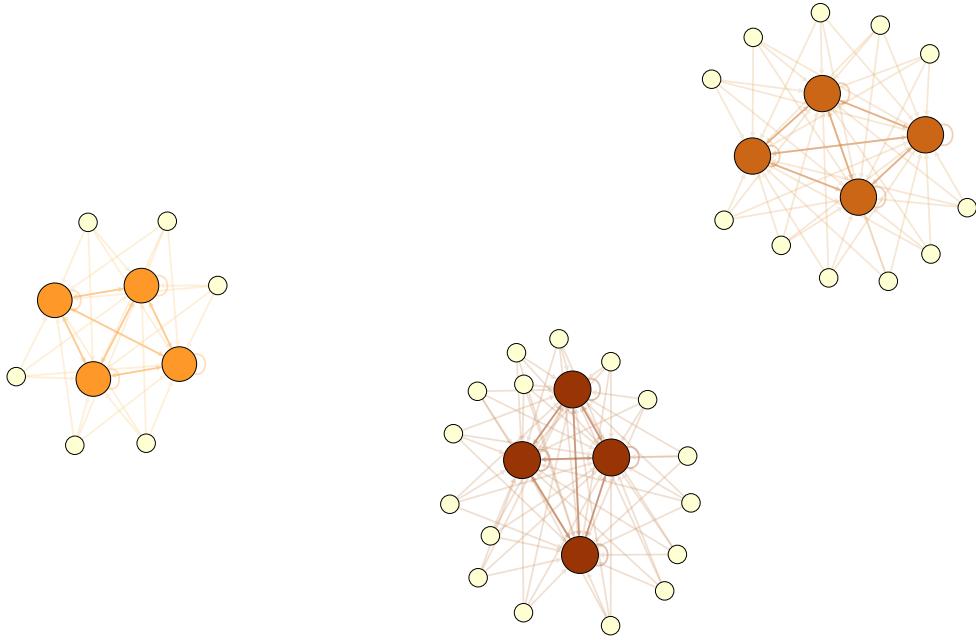


Figura 4.1: El sistema actual le da mucho peso a pocos hoteles y además no permite llegar de un grupo a otro.

ternativa es poner como arcos únicamente las 5 mejores recomendaciones de cada hotel, dado que serán los que los clientes verán directamente, sin tener que bajar la lista. Dado que no tenemos información sobre la importancia de cada posición en la lista, supondremos que son iguales, esto es, todos los arcos tendrán el mismo peso.

Utilizaremos el PageRank como la importancia de los nodos en lugar del grado de entrada. La razón de esto es que el grado de entrada da información puramente local. Es decir, si conocemos el grado de entrada, podemos saber cuántos hoteles le apuntan directamente a uno dado, pero no sabemos si hay una tendencia de la gráfica como conjunto a tender a esos nodos. Por eso utilizaremos mejor el PageRank para evaluar la importancia de los nodos, puesto que sí nos da información global.

## Análisis de la red

Dado que la gráfica resultante de las recomendaciones de los miles de hoteles de Best Day sería poco informativa en un entorno no interactivo, optamos por presentar los resultados del análisis en algunas zonas de interés. Primero haremos un análisis a una escala de varios destinos y posteriormente nos acercaremos a ver con más detalle uno solo.

Analizamos varios destinos importantes de México en la gráfica (4.2). En este caso, el tamaño de los nodos está dado por su PageRank, por lo que los nodos son más importantes mientras más grandes son, mientras que los colores representan el destino de cada hotel. Esta representación del sistema de recomendación transparenta las ventajas que tiene. En primer lugar, vemos que la importancia de los nodos está mucho más repartida que en la gráfica de ejemplo del sistema actual. Aunque hay algunos nodos grandes, no son los únicos importantes en la gráfica. Además de ello, parece ser que la gráfica está totalmente conectada (o casi) por destino; veremos esto a mayor detalle en una figura posterior. Otra ventaja es que es evidente que los destinos muy cercanos pueden estar conectados, como en el caso de los que están en Quintana Roo. Es interesante notar que si hay varios destinos en cadena, se puede llegar a lugares muy lejanos si se explora lo suficiente. Incluso se puede llegar de Cancún a Tulum, incluso a pesar de que están a 130 km de distancia. Esto puede ser una ventaja porque un cliente que no tiene un destino decidido puede realmente explorar alternativas similares a lo que se le había ocurrido inicialmente.

La gráfica (4.2) nos da una idea de lo que sucede con el algoritmo a gran escala. Si ahora nos acercamos a un destino en particular, podremos descubrir otras cosas interesantes. Dado que Cancún es el destino principal de Best Day, ahí es donde enfocaremos nuestro análisis. La figura (4.3) ilustra qué hoteles son los más importantes en esta zona. En este caso, el tamaño representa el PageRank de los hoteles, por lo que un hotel con un nodo grande tendrá mayor probabilidad de ser elegido al azar utilizando únicamente las recomendaciones.

Ya con una idea de qué hoteles son los más relevantes podemos hacer otro tipo de análisis. Las figuras (4.4) y (4.5) también incluyen únicamente hoteles en Cancún y el tamaño de los nodos está dado de la misma manera. Sin embargo, en estos casos el color corresponde al número de pasos que se requiere para llegar de un hotel determinado a los demás. En particular, el hotel más oscuro es el de origen y el color se va degradando conforme más

y más pasos se requiera para llegar a los demás. Los hoteles en gris no son accesibles desde el hotel original mediante las primeras cinco recomendaciones de los hoteles. Cabe notar que en realidad son cuatro recomendaciones efectivas, puesto que cada hotel aparece en el primer lugar de su propia lista.

Como podemos ver en la gráfica (4.4), de un hotel citadino y de negocios como La Quinta Inn and Suites podemos llegar sin problemas tanto a otros hoteles similares como de plano a hoteles populares en la zona si damos *click* en las recomendaciones adecuadas. Similarmente, la gráfica (4.5) muestra las posibilidades que se tiene empezando en el hotel Iberostar Cancún. Es claro ver que en el primer caso hay una mayor cantidad de hoteles accesibles que en el segundo, al menos mediante las primeras cinco recomendaciones. Esto se debe a que usualmente es más fácil explorar las regiones remotas de la red empezando en un hotel poco reconocido que en uno popular: la red tiene cierta tendencia a llevarnos a los hoteles populares.

## Conclusiones del análisis de redes

Si bien el sistema que construimos tiene muchas bondades, hay que tener algunas cosas en mente. Para ello, concluiremos nuestro análisis de redes con los siguientes puntos clave.

Antes que nada, hay que notar que si bien el sistema de recomendación fue construido de manera inteligente, no todas sus características están garantizadas. Por ejemplo, no hay razón alguna para pensar que todos los hoteles aparecerán en las recomendaciones de algún otro hotel. Es decir, es posible que nadie recomiende a un hotel en particular. Habrá que tener cuidado con esto y monitorear que no afecte el desempeño del sistema, aunque creemos que en la práctica sería difícil que esto representara un problema.

Otro aspecto que hay que tener en cuenta es que, aunque el precio no está restringido por abajo, la estructura de la red tiene cierta tendencia a llevar al cliente a los hoteles con PageRank alto. Habrá que observar el cambio en el comportamiento de los clientes para determinar si esto afecta negativamente a los hoteles menos comunes. Nuestra principal preocupación es que demasiados acaben en las mismas opciones siempre, aunque dudamos que el problema sea ni cercanamente tan grave como con el sistema actual.

Finalmente, en las redes anteriores hicimos dos grandes supuestos que no necesariamente se cumplirán en la práctica: (1) que los clientes sólo ven las primeras cinco recomendaciones (o sea cuatro distintas al original), y (2) que los clientes eligen al azar y con la misma probabilidad de cada hotel. El

primer supuesto no se cumplirá en la práctica porque los clientes tendrán más alternativas si ven más elementos de la lista de recomendaciones. Este hecho es ventajoso para el sistema porque ayudará, al menos en parte, a que haya menos hoteles que ningún otro recomiende. El segundo supuesto tampoco se cumplirá porque hay otros factores que motivan a un cliente a seleccionar una recomendación, como por ejemplo el precio, la foto (*thumbnail*) que se le presenta en la página o incluso el nombre del hotel. Se podría hacer un análisis exhaustivo con las herramientas de la próxima sección, pero no estamos seguros de que el impacto sería lo suficientemente importante para justificar dicho análisis.

## 4.2. Desempeño de la página

Google Analytics

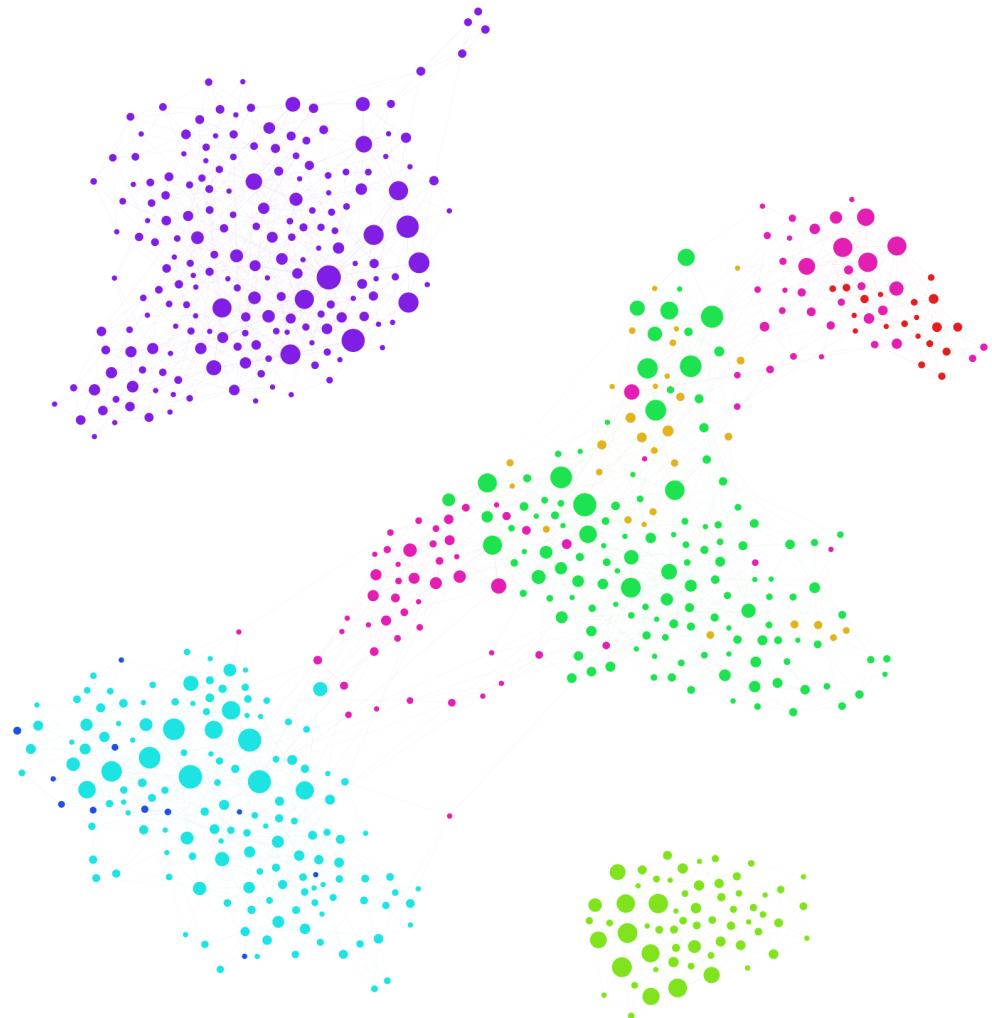


Figura 4.2: Red para diversos destinos: Ciudad de México (azul-morado), Cancún (turquesa), Playa del Carmen (verde), Riviera Maya (fucsia), Isla Mujeres (azul), Cozumel (naranja), Tulum (rojo) y Cuernavaca (verde-amarillo). El tamaño de los nodos está dado por el PageRank.

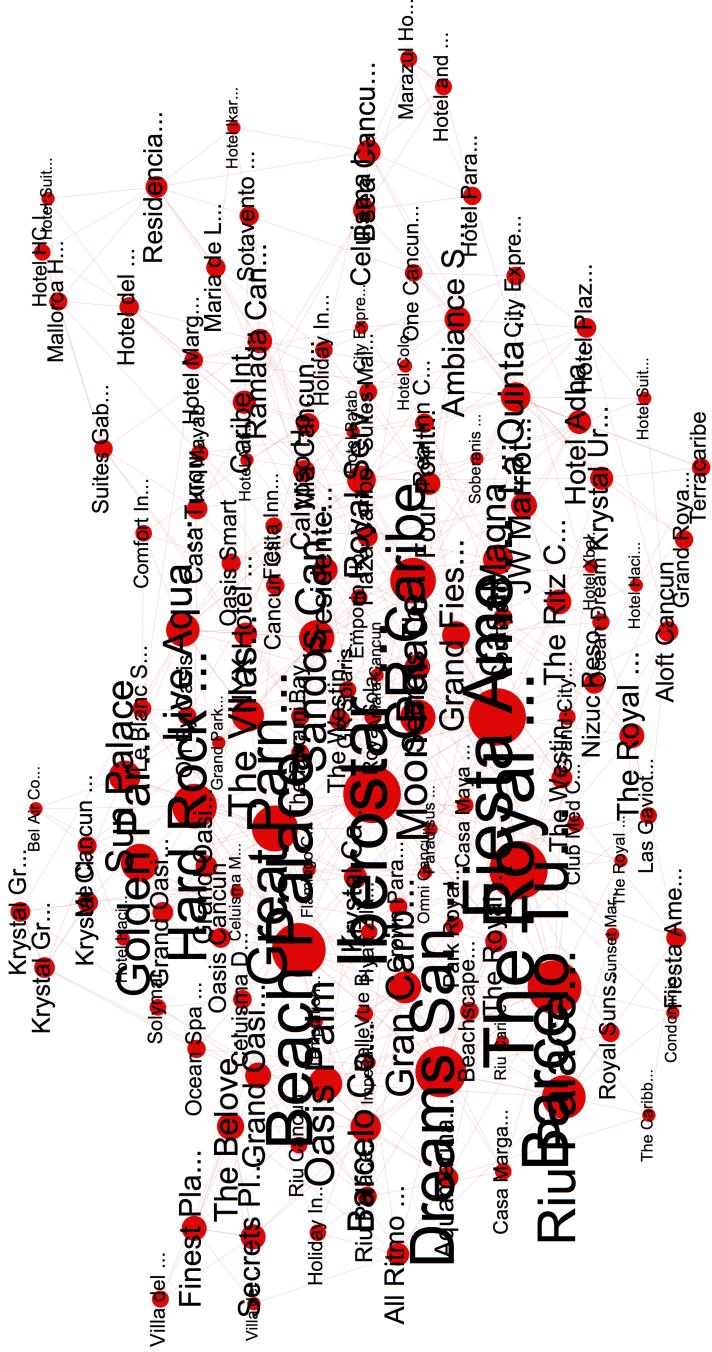


Figura 4.3: Hoteles en Cancún. La importancia (PageRank) está dada por el tamaño de los nodos. Sin contar los nodos inaccesibles, el diámetro de esta gráfica es de 7 pasos y la distancia promedio entre nodos es de 3.3.

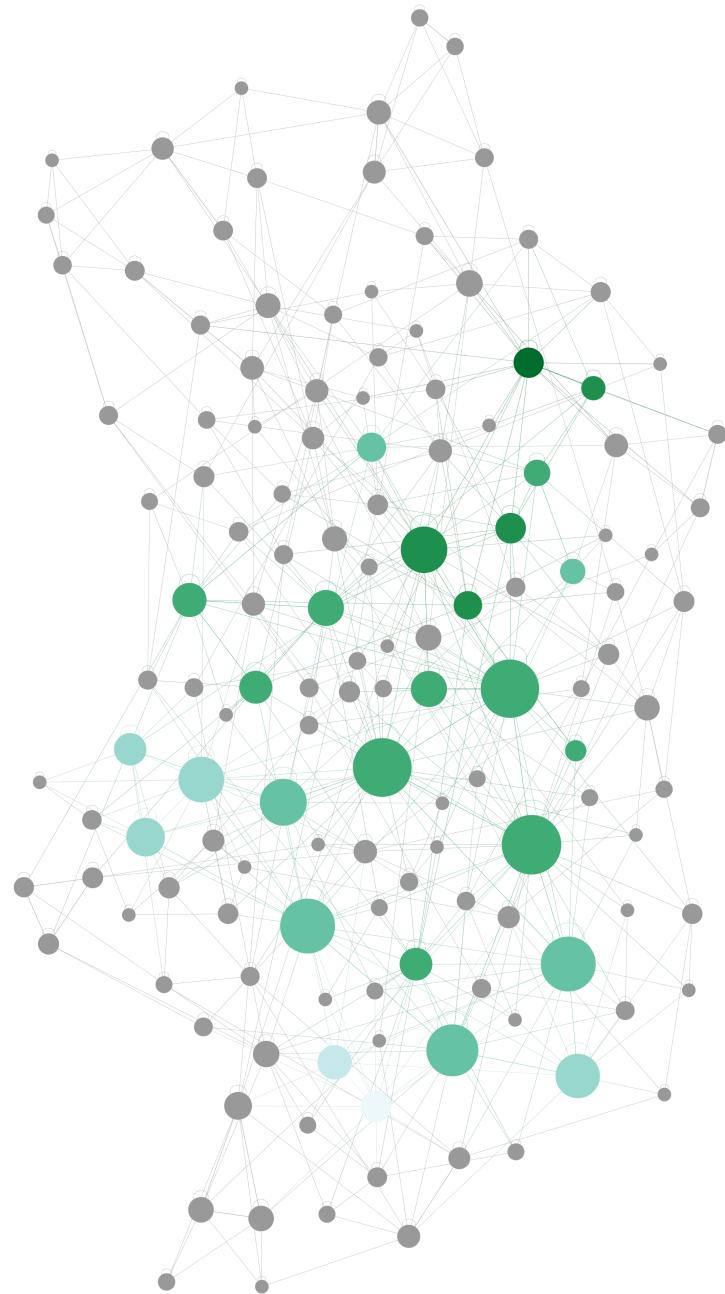


Figura 4.4: Tamaño: PageRank. Color: Más oscuro implica mayor cercanía (en recomendaciones) al hotel La Quinta Inn and Suites.

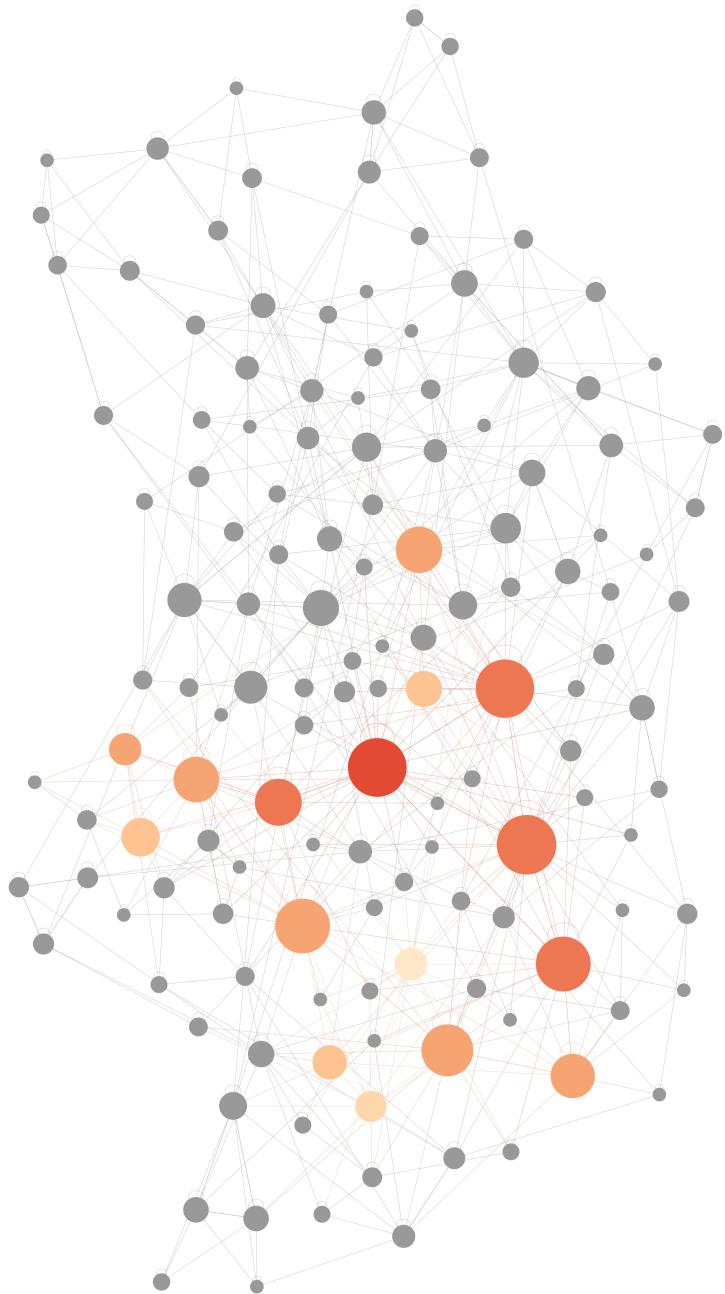


Figura 4.5: Tamaño: PageRank. Color: Más oscuro implica mayor cercanía (en recomendaciones) al hotel Iberostar Cancún.

# Capítulo 5

## Conclusiones

### 5.1. Trabajo futuro

Incorporar los hoteles vistos recientemente de forma dinámica. Porcentaje dinámico en forma de ‘U’ para el tope de precio.

Otras métricas

### 5.2. Comentarios finales

Por qué no distancia absoluta?

No es personalizado, sino de hoteles similares.

[Ver presentación para qué sí/no es o podría ser]

# Bibliografía

- [1] The R Project for Statistical Computing. <https://www.r-project.org/>, 2015.
- [2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [3] Google Inc. Google Analytics. <https://www.google.com/analytics/>, 2015.
- [4] Microsoft. SQL Server. <http://www.microsoft.com/en-us/server-cloud/products/sql-server/>, 2015.
- [5] Fionn Murtagh and Pierre Legendre. Ward’s hierarchical clustering method: Clustering criterion and agglomerative algorithm. 2011.
- [6] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.
- [7] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.
- [8] Cosma Shalizi. Distances between Clustering, Hierarchical clustering. notes from “statistics 36-350: Data mining”. 2009.