Pontificia Universidad Católica de Chile

Escuela de Ingeniería

Departamento de Computación

Sistemas Recomendadores IIC-3633

Propuesta

Francisco Pérez Páez

Francisco Rencoret Domínguez

23 de Septiembre, 2018

Contexto del Problema

Los viajes han acompañado a la humanidad desde sus comienzos. Los motivos son variados, pero van desde viajes por placer, por trabajo u otros. La cantidad de tareas que se deben realizar al planear un viaje son muchísimas y dentro de estas está la elección de los hoteles donde hospedarse. Equivocarse al momento de reservar un hotel puede reducir de manera significativa la calidad de un viaje.

Lo anterior habla de los problemas que afectan a los demandantes de hoteles, sin embargo, este problema es uno en el que las empresas de reservas masivas de hoteles pueden ayudar con un sistema recomendador. La idea es que dado las fechas e intereses del usuario, las empresas de reservas masivas sean capaces de recomendarles hoteles que le gusten al usuario para que ellos hagan reservas y puedan disfrutar de una buena experiencia en el lugar.

Problema y Justificación

El problema de recomendación de un hotel a un cliente es crucial en el modelo de negocios de empresas como Booking o Expedia, ya que se basan en recomendar hoteles a sus usuarios con el objetivo de que su estadía sea acorde a sus expectativas.

Dada la gran variedad de hoteles que se ofrecen, la tarea de encontrar el hotel que el usuario realmente quiere sin haber estado nunca ahí es una tarea cada vez más difícil. Es por esto que el problema de recomendación de un hotel a un potencial cliente es uno muy importante de resolver. Además, dado el aumento de tecnología y la facilidad para viajar, Expedia recibe altas cantidades de visitas diarias, por lo que una mejora en la recomendación de hoteles puede significar en una alza de ingresos importante para ellos.

Datos

Buscando en Kaggle, encontramos un *dataset* provisto por Expedia que contiene una muestra aleatoria de datos contextuales de la reserva hecha por los clientes y el cluster de hoteles que eligieron en ese momento.

Estos datos contextuales son variados pero incluyen el momento de la reserva; días que se pretende estar hospedado; lugar al que se quiere ir; búsquedas pasadas de otros hoteles; hora y lugar en la que se reservó; si es que lo hizo desde un móvil o no, entre otros. Por otro lado, el cluster de hoteles es un identificador de un cluster en que se agrupan hoteles similares basado en precio histórico; ratings de usuarios; ubicación geográfica relativa al centro de una ciudad, etc. Expedia no presenta los hoteles de los clusters.

Objetivos

Hay tres principales objetivos que esta investigación pretende cumplir.

En primer lugar, generar un sistema recomendador que, basado en datos contextuales de búsqueda de hoteles, prediga el identificador del cluster de hoteles al que el usuario prefiera reservar.

En segundo lugar, utilizar conocimientos de Deep Learning para realizar una recomendación que supere a métodos más simples como los basados en memoria.

En tercer lugar, se aspira a que la recomendación tenga un rendimiento sea comparable con los resultados obtenidos por otros equipos que participaron en la competencia de Kaggle.

Solución propuesta

Como fue mencionado anteriormente, el *dataset* provisto por Expedia presenta una secuencia de reservas de hoteles por parte de los usuarios. Cada reserva tiene asociado un *timestamp* junto con datos contextuales acerca de la reserva. Considerando la naturaleza secuencial de los datos, creemos que aplicar un modelo de Red Recurrente RNN haría sentido.

Nuestra intuición nos dice que existe una correlación en la data dada su secuencia, por ejemplo, si una persona va dos veces al caribe en la misma fecha en dos años es probable que vaya de nuevo el año que le sigue. Este motivo nos hace creer que una RNN podrá captar esta correlación de la secuencia y lograr predecir el siguiente destino del usuario dado los datos contextuales.

Decidimos optar por un alcance desde el aprendizaje profundo para resolver este problema porque sabemos que es un rubro en cual se ha avanzado mucho; obteniendo los mejores resultados para problemas como este. Notamos que el problema de recomendar el siguiente hotel al usuario es un problema común y corriente de recurrencia, donde dado información de contexto, se predice cuál es la probabilidad de que el usuario escoja un hotel de todos los clusters posibles.

Consecuentemente, para crear las recomendaciones vamos a priorizar los clusters y retornar una lista de los clusters más probables de ser reservados. Como no tenemos acceso a los hoteles dentro de los clusters, no podemos generar recomendaciones sobre los hoteles mismos, pero creemos que recomendando clusters con su probabilidad de reserva agregaría valor.

Considerando lo anterior, vamos a aplicar un modelo RNN para ajustar la secuencia y planeamos explorar técnicas del estado del arte para aquellas redes. Los modelos de

atención sobre la recurrencia han demostrado aumentar las capacidades de generalización de los modelos, por lo que planeamos explorar ese rubro para ver su impacto.

Descripción de experimentos

Dado que el dataset ya viene separado en *train* y *test*, vamos a mantener esa estructura para nuestro entrenamiento y evaluación. Haremos esto para apegarnos al *challenge* propuesto por Expedia, para poder luego comparar justamente nuestros resultados con los propuestos en Kaggle.

En cuanto a los modelos, vamos a seguir el principio de Occam's Razor y vamos a intentar resolver este problema con un modelo simple. Crearemos un modelo básico de RNN donde comenzaremos haciendo secuencias de largo igual al máximo de reservas hechas por un usuario, agregando *paddings* cuando sea necesario. Vamos a comenzar con un estado oculto de 128 (típicamente usado en problemas así) pero iteraremos sobre esta dimensión.

Si eso nos trae buenos resultados, intentaremos aplicar recurrencias de largo variable (para evitar el *padding*) e implementar modelos de atención. Usaremos atención sobre las celdas de recurrencia, donde usaremos una red Feed-Forward para calcular los coeficientes de atención y así el posterior vector ponderado de atención. Usaremos Keras para la implementación, pero si el modelo de atención nos trae muchos problemas intentaremos con PyTorch.

En cuanto al entrenamiento, vamos a usar nuestro set de test como set de validación para crear un *Callback* de *EarlyStopping* y así evitar que el modelo se sobreajuste a los datos. Además, dependiendo del tiempo de entrenamiento, haremos pruebas con distintos hiper-parámetros para optimizar el aprendizaje. Usaremos Adam (optimizador más usado hoy en día) pero probaremos distintos learning rates y *thresholds* para el *EarlyStopping*.

Dado que estamos evaluando la capacidad de predecir a qué cluster de hotel el usuario debería hacer la reserva, vamos a usar una función de pérdida *categorical cross entropy* para evaluar si el modelo escoge el cluster correctamente. Por otra parte, utilizaremos métricas de accuracy, precision, recall y F1-Score para poder evaluar nuestros experimentos. Cuando hayamos obtenido la configuración óptima del modelo y entrenamiento, vamos a comparar nuestros resultados sobre el set de test con los otros resultados aplicados en Kaggle.

Bibliografía

1. Dataset: https://www.kaggle.com/c/expedia-hotel-recommendations/data
2. Redes Neuronales Recurrentes en Sistemas Recomedadores: https://cs224d.stanford.edu/reports/LiuSingh.pdf
3. Redes Neuronales Profundas y Sistemas hybridos de recomendación: https://arxiv.org/pdf/1707.07435.pdf
4. Modelos de Atención para RNNs: https://arxiv.org/pdf/1409.0473.pdf