
Learning to Rank

Alondra Berzunza

En la práctica, los sistemas de recomendación solo muestran al usuario los elementos como una lista ordenada.

Los usuarios usualmente le ponen atención a los primeros elementos de la lista.

Los modelos basados en predecir la calificación no consideran explícitamente el ordenamiento.

Usualmente primero se realiza la predicción de la calificación de cada elemento y posteriormente se ordena a partir de esta calificación.

Learning to Rank

Utiliza un conjunto de técnicas de machine learning (aprendizaje automático) diseñadas para entrenar un modelo que ordene o clasifique una lista de elementos según su relevancia o preferencia. A diferencia de los modelos de clasificación o regresión que predicen un valor único, los algoritmos LTR predicen el orden óptimo de un conjunto de elementos.

¿Cómo Funciona?

Datos de entrenamiento: El proceso requiere un conjunto de datos de entrenamiento que contenga listas de elementos (por ejemplo, documentos o productos) emparejadas con una calificación de relevancia. Esta calificación puede ser:

- Numérica u ordinal: Un valor que indica qué tan relevante es un elemento.
- Binaria: Un juicio simple de "relevante" o "no relevante".
- Interacciones del usuario: Datos sobre el comportamiento del usuario, como clics, compras o tiempo de visualización, que señalan la relevancia percibida.

¿Cómo Funciona?

Entrenamiento del modelo: Con estos datos, se entrena un modelo de machine learning para aprender una función de puntuación. Esta función asigna una puntuación de relevancia a cada elemento en relación con una consulta o contexto específico.

Clasificación de nuevos datos: Una vez entrenado, el modelo utiliza esta función para clasificar listas de nuevos elementos que no ha visto antes. Ordena los elementos según sus puntuaciones de relevancia, presentando los más relevantes primero.

Learning to Rank

En learning to rank, se aplica aprendizaje supervisado para realizar el ordenamiento de un conjunto de elementos.

Enfoques:

- A nivel elemento: cada elemento se clasifica como relevante o no relevante.
 - A nivel par: predice el orden relativo de cada par de elementos.
 - A nivel lista: genera el orden de una lista completa.
-

Nivel Elemento (Pointwise)

Se basa en los atributos de cada elemento de forma individual.

- Datos de entrenamiento: elementos y su calificación (e.g. relevante/no relevante o valor numérico)
- Ejemplos: Pranking o McRank

Trata la tarea de clasificación como un problema de regresión o clasificación a nivel de documento individual. El modelo predice la puntuación de relevancia de un documento sin considerar la relación con otros.

Nivel Par (Pairwise)

Compara pares de documentos para determinar cuál debería tener una clasificación superior. El modelo se centra en minimizar los errores en las preferencias de ordenación entre pares de documentos.

Aprende un modelo h que determina el orden relativo de dos elementos $x(i)$, $x(j)$, es decir, si $x(i)$ va antes o después que $x(j)$.

- Datos de entrenamiento: pares de elementos y cuál es más relevante.
- Ejemplo: RankNet, LambdaRank o RankingSVM

Nivel Lista (listwise)

Optimiza una función de pérdida basada en el ordenamiento completo de una lista.

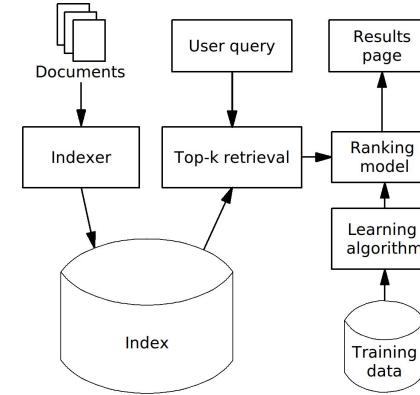
Ejemplos de funciones de pérdida: normalized cumulative discounted gain (NDCG) y mean reciprocal rank (MRR)

Problema: hay muchos posibles ordenamientos de una misma lista.

- Ejemplos: ListNet, ListMLE o PermuRank

LTR y Recuperación de la información

Aplicado a motores de búsqueda donde hay un conjunto de consultas Q, cada uno con su correspondiente conjunto de respuesta S y el problema es ordenar S.



RankNet

Método de learning to rank en el que se entrena un modelo optimizando una función diferenciable respecto a sus parámetros.

Usualmente una red neuronal, aunque también se han usado árboles de potenciación del gradiente.

Cada ejemplo $x(i)$ se mapea a un número $h(x(i))$

Ej. red neuronal tipo siamesa entrenada para aprender el orden relativo de dos elementos.

Entrenamiento

Estas salidas se mapean a una probabilidad de que $x(i)$ es más relevante que $x(j)$

donde σ es un parámetro que determina la forma de la función sigmoide.

- Se busca minimizar la entropía cruzada binaria.

LambdaRank

Es similar a RankNet pero en el entrenamiento se busca maximizar el Normalized Discounted Cumulative Gain (NDCG).

donde T es el nivel de truncamiento, y (i) es la etiqueta asociada al i -ésimo elemento de la lista.

Se especifican los gradientes directamente, en lugar de derivarlos.