IIC3633 - 2020
Sistemas recomendadores

Self-Supervised Reinforcement Learning for Recommender Systems (2020)

- Xin Xin
- Ioannis Arapakis
- Alexandros Karatzoglou
- Joemon M. Jose

Explicado por: Vicente Díaz y Martín Vinay

Tabla de contenido

- Contexto
- Marco teórico
- Trabajo relacionado
- Modelo propuesto
- Experimentos
- Principales resultados
- Contribuciones

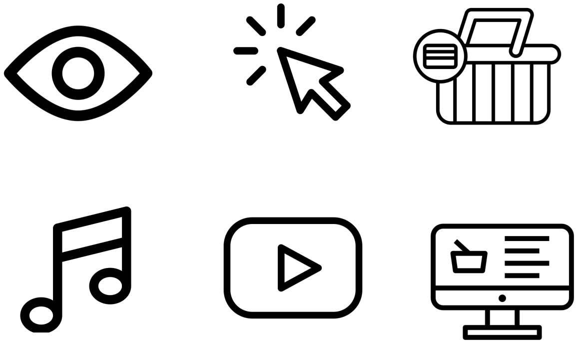
Contexto

Recomendación secuencial

- Su objetivo es generar la recomendación del siguiente ítem a partir de una secuencia de interacciones usuario-ítem.

- Es uno de los casos más comunes dentro de los sistemas recomendadores.

- Entrenamiento *self-supervised*: aprende a predecir el mismo tipo de data.



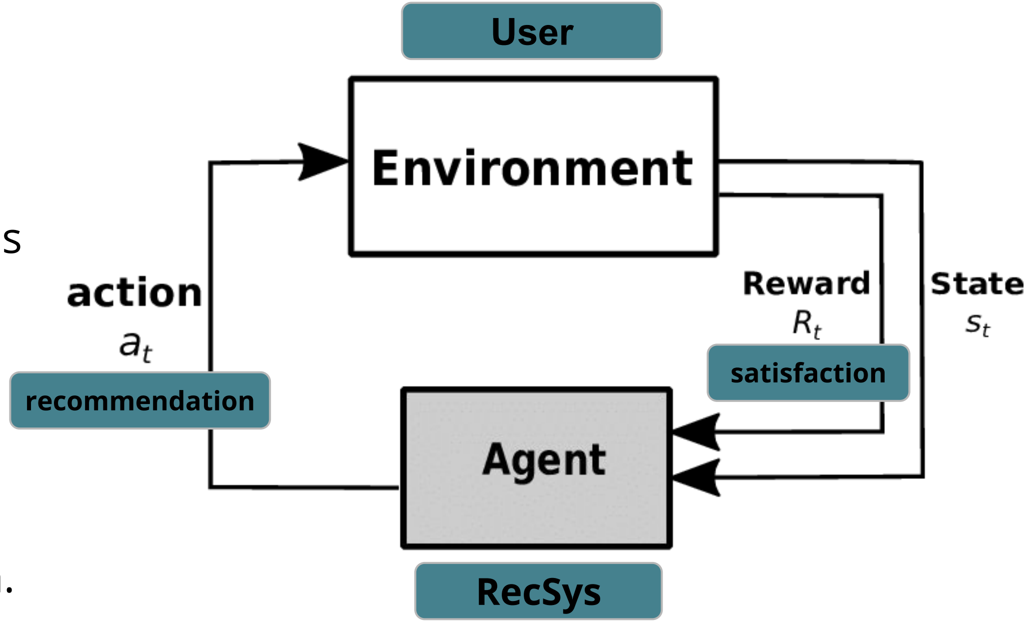
$$x_{1:t} = \{x_1, x_2, \dots, x_{t-1}, x_t\} \rightarrow ¿x_{t+1}?$$

Recomendación secuencial

- Problemas:
  - Este tipo de *approach* tiende a ser subóptimo, al solo hacer un *fit* entre las predicciones y las señales de supervisión.
  - Interacciones son de variados tipos (ej. clicks y compras), información comúnmente no incluida en estos modelos.

Reinforcement learning recommendation

- Ventajas:
  - Maximiza satisfacción a largo plazo.
  - Modelos con objetivos flexibles (ej. diversidad y novedad)
- Problemas:
  - Política aprendida en entrenamiento offline no es óptima al no haber exploración.
  - Falta de datos y recompensas negativas



Marco teórico

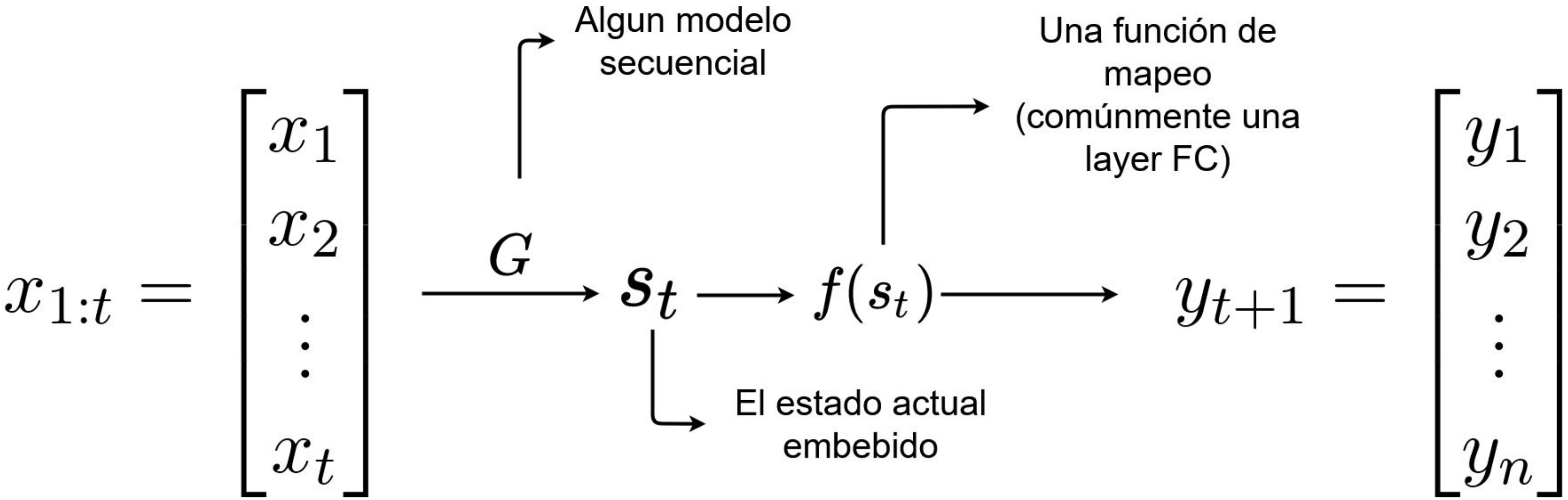
Next item recommendation

Interacción en i

Top-n recomendaciones

$$x_{1:t} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} \longrightarrow y_{t+1} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Next item recommendation



Reinforcement Learning (MDP)

$\mathcal{A}$: Espacio discreto de acciones t.q. $a_t = x_{t+1} \in \mathcal{A}$

$\mathcal{S}$: Espacio continuo de estados t.q. $s_t = G(x_{1:t}) \in \mathcal{S}$

$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ es la probabilidad de transicion de estados

$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ es una funcion de premio/*reward*

γ : factor de descuento para recompensa futura

Se busca una politica $\pi_\theta(a|s)$ que maximice el reward dado un estado $s \in \mathcal{S}$ sobre las posibles acciones $\mathcal{A}$

Q-Learning

reward instantáneo basado en política de recompesa diseñada

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \, max_{a'} Q(s_{t+1}, a')$$

Función símil a la política $\pi_\theta(a|s)$ a aprender

Esperanza de *reward* futuro

Ecuación de Bellman

Trabajo relacionado

Algoritmos recomendación secuencial

- Primeros modelos:
  - Cadenas de Markov: problema en modelar señales de secuencias complejas.
  - Factorización matricial: no modela el orden de las interacciones user-item
- **Estado de arte (deep learning)**:
  - Gated recurrent units (GRU)
  - Basados en CNN: Caser y NItNet
  - Self-attention with Transformers: SASRec

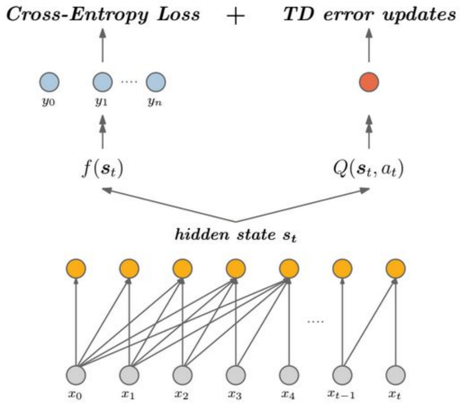
Modelo propuesto

Frameworks

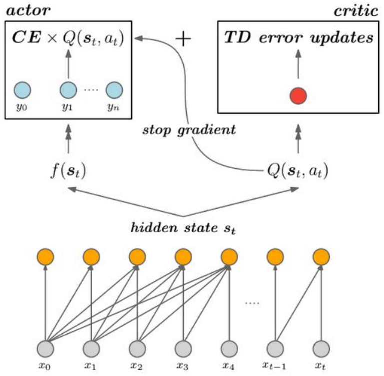
Los autores presentan dos modelos:

- **SQN** (Self-supervised Q learning) : A partir de los estados embebidos se extiende un *head self-supervised* y un *head* de RL. La función de pérdida de estas cabezas es compartida. La recomendación final es a partir de la *self-supervised head.*

- **SAC** (Self-supervised Actor-Critic) : Similar al modelo SQN, pero el *head* RL actúa como crítico sobre el actor (*head self-supervised)*.

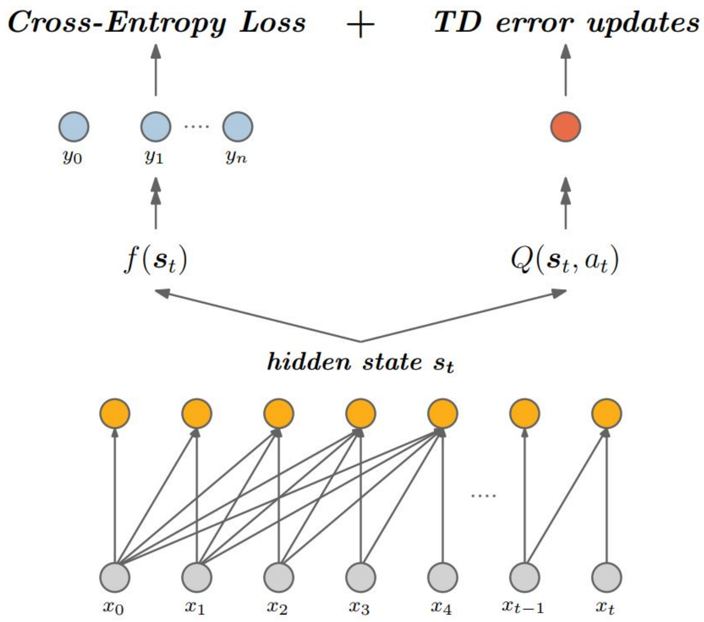


(b) SQN architecture.



(c) SAC architecture. CE stands for cross-entropy.

SQN: Self-supervised Q learning



(b) SQN architecture.

$$L_{SQN} = L_q + L_s$$

Loss del head supervisado

Indicator function

$$L_s = -\sum_{i=1}^{n} Y_i log(p_i)$$

donde $p_i = \frac{e^{y_i}}{\sum_{i'=1}^{n} e^{y_{i'}}}$

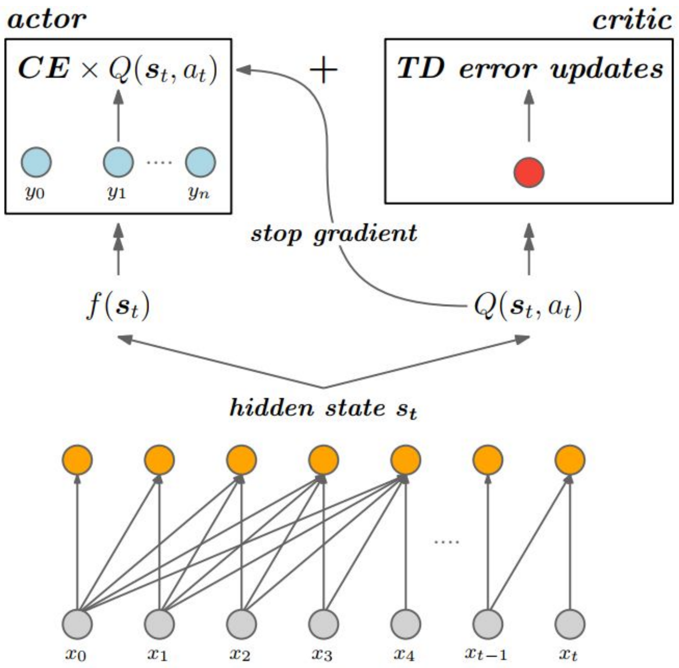
Loss del *head* RL

$$\sqrt{L_q} = r(s_t, a_t) + \gamma\, max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)$$

donde $Q(s_t, a_t) = \sigma(s_t h_t^T + b)$,

h_t y b siendo pesos y bias de una layer FC

SAC: Self-supervised Actor-Critic



(c) SAC architecture. CE stands for cross-entropy.

$$L_{SAC} = L_a + L_s$$

Loss del agente

$$L_a = L_s \times Q(s_t, a_t)$$

Acciones con alto valor de Q, deberían recibir pesos incrementados en la *self-supervised head,* y viceversa

Experimentos realizados

Datasets

Table 1: Dataset statistics.

Dataset	RC15	RetailRocket
#sequences	200,000	195,523
#items	26,702	70,852
#clicks	1,110,965	1,176,680
#purchase	43,946	57,269

YOOCHOOSE Personalization Solutions

retailrocket

- Ambas empresas prestan servicios en el área de e-commerce.

Baselines

Modelos del estado del arte para recomendación secuencial regenerativa:

GRU

NItNet

Caser

SASRec

Research questions

RQ1

How do the proposed methods **perform** when **integrated** with existing recommendation models?

RQ2

How does the **RL** component **affect performance**, including the reward setting and the discount factor?

RQ3

What is the performance if we **only use Q-learning** for recommendation?

Principales resultados

RQ1: Model performance

Table 2: Top-k recommendation performance comparison of different models ($k = 5, 10, 20$) on RC15 dataset. NG is short for NDCG. Boldface denotes the highest score. $*$ denotes the significance p-value < 0.01 compared with the corresponding baseline.

Models	purchase						click					
	HR@5	NG@5	HR@10	NG@10	HR@20	NG@20	HR@5	NG@5	HR@10	NG@10	HR@20	NG@20
GRU	0.3994	0.2824	0.5183	0.3204	0.6067	0.3429	0.2876	0.1982	0.3793	0.2279	0.4581	0.2478
GRU-SQN	0.4228*	0.3016*	0.5333*	0.3376*	0.6233*	0.3605*	**0.3020***	**0.2093***	**0.3946***	**0.2394***	**0.4741***	**0.2587***
GRU-SAC	**0.4394***	**0.3154***	**0.5525***	**0.3521***	**0.6378***	**0.3739***	0.2863	0.1985	0.3764	0.2277	0.4541	0.2474
Caser	0.4475	0.3211	0.5559	0.3565	0.6393	0.3775	0.2728	0.1896	0.3593	0.2177	0.4371	0.2372
Caser-SQN	0.4553*	0.3302*	0.5637*	0.3653*	0.6417*	0.3862*	**0.2742**	**0.1909**	**0.3613**	**0.2192**	**0.4381**	**0.2386**
Caser-SAC	**0.4866***	**0.3527***	**0.5914***	**0.3868***	**0.6689***	**0.4065***	0.2726	0.1894	0.3580	0.2171	0.4340	0.2362
NItNet	0.3632	0.2547	0.4716	0.2900	0.5558	0.3114	0.2950	0.2030	0.3885	0.2332	0.4684	0.2535
NItNet-SQN	0.3845*	0.2736*	0.4945*	0.3094*	**0.5766***	0.3302*	**0.3091***	**0.2137***	**0.4037***	**0.2442***	**0.4835***	**0.2645***
NItNet-SAC	**0.3914***	**0.2813***	**0.4964***	**0.3155***	0.5763*	**0.3357***	0.2977*	0.2055*	0.3906	0.2357*	0.4693	0.2557*
SASRec	0.4228	0.2938	0.5418	0.3326	0.6329	0.3558	0.3187	0.2200	0.4164	0.2515	0.4974	0.2720
SASRec-SQN	0.4336	0.3067*	0.5505	0.3435*	0.6442*	0.3674*	**0.3272***	**0.2263***	**0.4255***	**0.2580***	**0.5066***	**0.2786***
SASRec-SAC	**0.4540***	**0.3246***	**0.5701***	**0.3623***	**0.6576***	**0.3846***	0.3130	0.2161	0.4114	0.2480	0.4945	0.2691

RQ1: Model performance

Table 2: Top-k recommendation performance comparison of different models ($k = 5, 10, 20$) on RC15 dataset. NG is short for NDCG. Boldface denotes the highest score. $*$ denotes the significance p-value < 0.01 compared with the corresponding baseline.

Models	purchase						click					
	HR@5	NG@5	HR@10	NG@10	HR@20	NG@20	HR@5	NG@5	HR@10	NG@10	HR@20	NG@20
GRU	0.3994	0.2824	0.5183	0.3204	0.6067	0.3429	0.2876	0.1982	0.3793	0.2279	0.4581	0.2478
GRU-SQN	0.4228*	0.3016*	0.5333*	0.3376*	0.6233*	0.3605*	**0.3020***	**0.2093***	**0.3946***	**0.2394***	**0.4741***	**0.2587***
GRU-SAC	**0.4394***	**0.3154***	**0.5525***	**0.3521***	**0.6378***	**0.3739***	0.2863	0.1985	0.3764	0.2277	0.4541	0.2474
Caser	0.4475	0.3211	0.5559	0.3565	0.6393	0.3775	0.2728	0.1896	0.3593	0.2177	0.4371	0.2372
Caser-SQN	0.4553*	0.3302*	0.5637*	0.3653*	0.6417*	0.3862*	**0.2742**	**0.1909**	**0.3613**	**0.2192**	**0.4381**	**0.2386**
Caser-SAC	**0.4866***	**0.3527***	**0.5914***	**0.3868***	**0.6689***	**0.4065***	0.2726	0.1894	0.3580	0.2171	0.4340	0.2362
NItNet	0.3632	0.2547	0.4716	0.2900	0.5558	0.3114	0.2950	0.2030	0.3885	0.2332	0.4684	0.2535
NItNet-SQN	0.3845*	0.2736*	0.4945*	0.3094*	**0.5766***	0.3302*	**0.3091***	**0.2137***	**0.4037***	**0.2442***	**0.4835***	**0.2645***
NItNet-SAC	**0.3914***	**0.2813***	**0.4964***	**0.3155***	0.5763*	**0.3357***	0.2977*	0.2055*	0.3906	0.2357*	0.4693	0.2557*
SASRec	0.4228	0.2938	0.5418	0.3326	0.6329	0.3558	0.3187	0.2200	0.4164	0.2515	0.4974	0.2720
SASRec-SQN	0.4336	0.3067*	0.5505	0.3435*	0.6442*	0.3674*	**0.3272***	**0.2263***	**0.4255***	**0.2580***	**0.5066***	**0.2786***
SASRec-SAC	**0.4540***	**0.3246***	**0.5701***	**0.3623***	**0.6576***	**0.3846***	0.3130	0.2161	0.4114	0.2480	0.4945	0.2691

RQ2: RL parameters

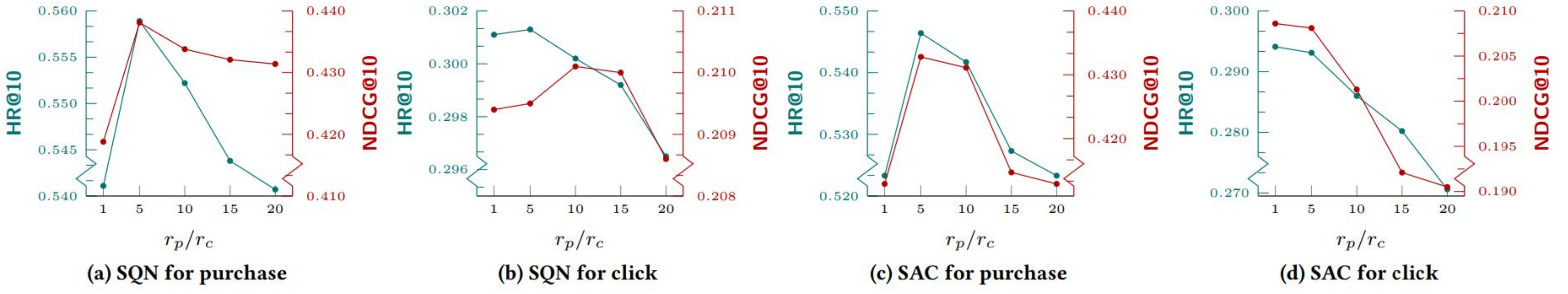
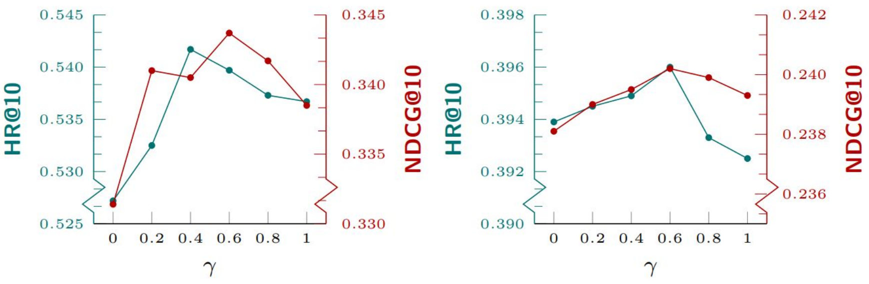


Figure 4: Effect of reward settings on RetailRocket

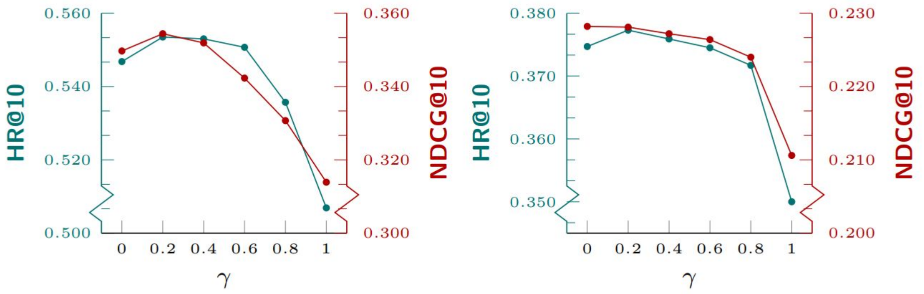
- Para la comparación de desempeño se eligió un r_p / r_c = 5.

RQ2: RL parameters



(a) SQN for purchase (b) SQN for click

Figure 5: SQN with different discount factors



(a) SAC for purchase (b) SAC for click

Figure 6: SAC with different discount factors

- Para la comparación de desempeño se eligió un γ = 0.5.

RQ3: Only Q-learning

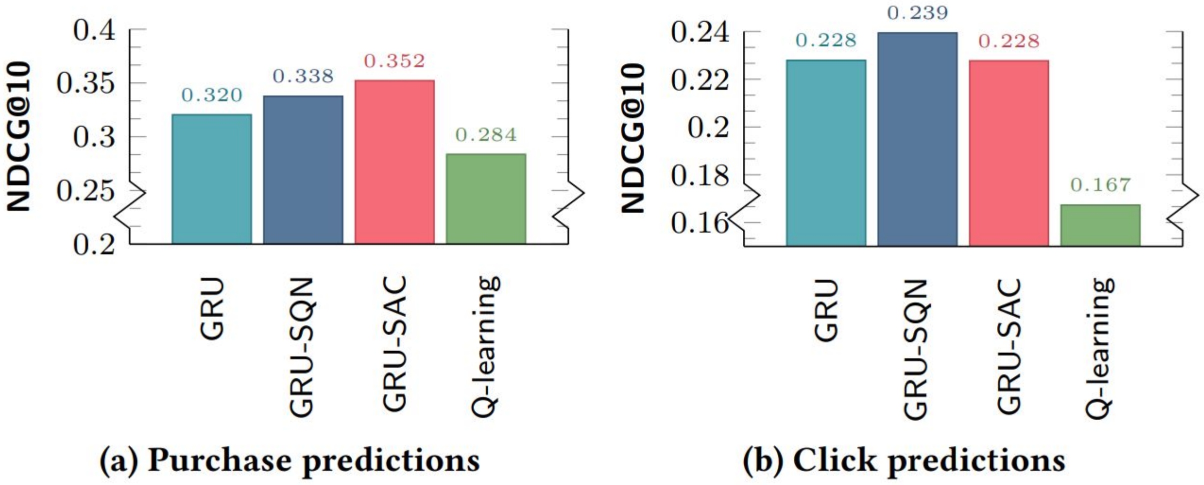


Figure 8: Comparison of NDCG when only using Q-learning for recommendations.

Contribuciones

Contribuciones

Las contribuciones de este *paper* son:

- Un modelo basado en *self-supervised reinforcement learning for sequential recommendation.*
- Dos frameworks (SQN y SAC) para co-entrenar la cabeza supervisada y la cabeza RL.
- Resultados experimentales efectivos en dos datasets del mundo real del e-commerce.

Referencias

- Hado V Hasselt. 2010. Double Q-learning. In Advances in Neural Information Processing Systems. 2613–262.
- Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management. ACM, 843–852.
- Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 197–206.

IIC3633 - 2020
Sistemas recomendadores

Self-Supervised Reinforcement Learning for Recommender Systems (2020)

- Xin Xin
- Ioannis Arapakis
- Alexandros Karatzoglou
- Joemon M. Jose

Explicado por: Vicente Díaz y Martín Vinay