

Collaborative Recurrent Neural Networks for Dynamic Recommender Systems

Ko et al[1]

Juan Pablo de Vicente

Sistemas Recomendadores

October 18, 2018

- Gran número de bienes y servicios online
- Mucha información puede ser agobiante para el usuario, por lo que esta se filtra y ordena mediante sistemas recomendadores
- Los sistemas pueden usar feedback explícito o implícito del usuario

- El comportamiento de los usuarios es altamente contextual
- La información explícita dada por el usuario generalmente solo se da una vez en el tiempo, por lo que no nos entrega mucha información contextual.

Feedback Implícito

- La información implícita recolectada es más abundante y fácil de recolectar
- Generalmente se recolecta en logs de actividades
- Este tipo de feedback se renueva con más frecuencia, por lo que modelos dinámicos pueden beneficiarse más de él
- Los activity logs son bastante convencionales para muchos sitios, por lo que el método es ampliamente aplicable

Gracias al aspecto temporal de los activity logs y a la abundancia de información del feedback implícito, podemos obtener una aproximación del comportamiento contextual del usuario.

- Set de usuarios U , donde u representa al usuario.
- Set de items I , donde i representa al usuario.
- Para cada usuario $u \in U$ tenemos una secuencia $x^{(u)} = [x_1^{(u)}, \dots, x_{T_u}^{(u)}]$ con elementos $x_t^{(u)} \in I$
- No estamos interesados en el tiempo absoluto, sino que en los patrones de tiempo, por lo que definimos $x_{<t}^{(u)} = [x_1^{(u)}, \dots, x_{t-1}^{(u)}]$ y $x_{\geq t}^{(u)} = [x_t^{(u)}, \dots, x_{T_u}^{(u)}]$
- De esta forma deberíamos poder cuantificar la probabilidad de que un usuario u consuma en $t=k, k+1, k+2, \dots$ teniendo acceso a $x_{<k}^{(u)}$

La propuesta de predicción es modelar las secuencias observadas de manera probabilística.

$$P_{\theta}(\mathbf{x}^{(u)}) = \prod_{t=1}^{T_u} P_{\theta}(x_t^{(u)} \mid \mathbf{x}_{<t}^{(u)})$$

$$P_{\theta}(x_t^{(u)} \mid \mathbf{x}_{<t}^{(u)}) = \mathbf{p}_t^{(u)}(\theta, \mathbf{x}_{<t}^{(u)})$$

Donde θ corresponde a parámetros del modelo y $\mathbf{p}_y^{(u)}$ es un vector de probabilidades.

Métrica de evaluación

Average negative log likelihood calculada a partir de un recorte de la secuencia de la forma: $x_{<r_u}^{(u)} x_{\geq r_u}^{(u)}$, $1 < r_u \leq T_u$

$$E(\mathbf{x}_{\geq r_1}^{(1)}, \dots, \mathbf{x}_{\geq r_U}^{(U)} \mid \boldsymbol{\theta}) = -\frac{1}{U} \sum_{u \in \mathcal{U}} \frac{1}{T_u - r_u + 1} \log P_{\boldsymbol{\theta}}(\mathbf{x}_{\geq r_u}^{(u)})$$

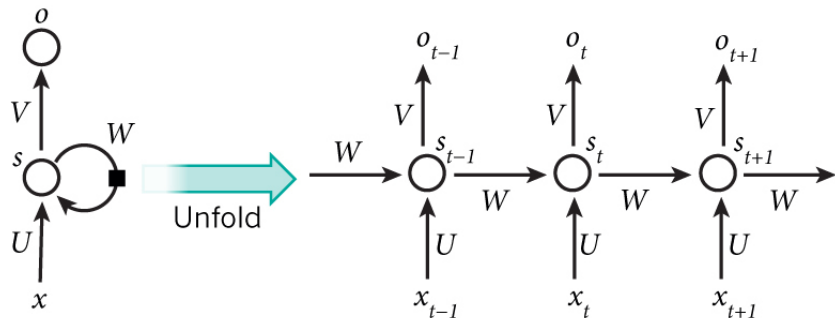
¿Qué es lo que se busca?

- Flexibilidad: Modelo capaz de dar buenos resultados en distintos dominios
- Dependencias de rango alto: Modelo suficientemente poderoso como para poder representar dependencias en secuencias largas
- Colaboración: Tener componentes que aprendan de todos los usuarios a la vez que componentes propias para cada usuario

RNN

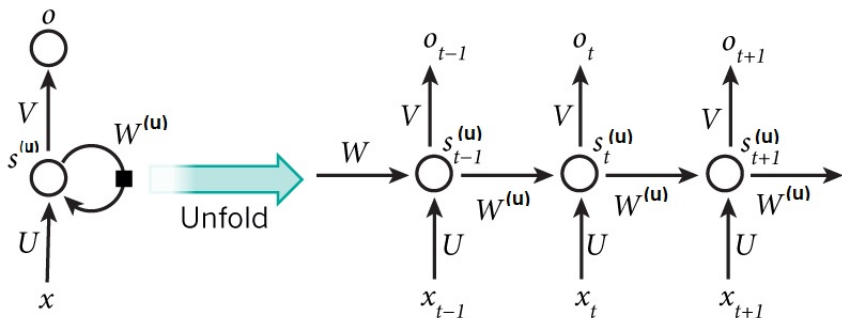
Un modelo basado en Recursive Neural Networks (RNN) puede hacer uso eficiente de la estructura secuencial multi-usuario del dataset aprovechando también el aspecto colaborativo.

¿Que es una RNN?



Collaborative-RNN (CRNN)

- Los pesos de entrada U y salida V se pueden considerar como embeddings a los valores reales, por lo que son comunes para todos los usuarios. De esta forma se ahorra en cantidad de parámetros.
- Los estados internos deben representar el contexto de cada usuario, por lo que cada usuario tendrá su propia matriz W



Static Uniform

$$P(x_t^{(u)} | \mathbf{x}_{<t}^{(u)}) = P(x_t^{(u)}) = \frac{1}{I}$$

Para I items.

n-grams models

¿Cual es la distribución de probabilidades del siguiente ítem conociendo los n anteriores?

$$c_i = \sum_{u \in \mathcal{U}} \sum_{t=1}^{T_u} \mathbb{1}(x_t^{(u)} = i), \quad b_{ij} = \sum_{u \in \mathcal{U}} \sum_{t=2}^{T_u} \mathbb{1}(x_t^{(u)} = i, x_{t-1}^{(u)} = j).$$

- Unigram

$$P(x_t^{(u)} = i | \mathbf{x}_{<t}^{(u)}) = P(x_t^{(u)} = i) = \frac{c_i + \epsilon}{\sum_{k \in \mathcal{I}} (c_k + \epsilon)}.$$

- Bigram

$$P(x_t^{(u)} = i | \mathbf{x}_{<t}^{(u)}) = P(x_t^{(u)} = i | x_{t-1}^{(u)} = j) = \frac{b_{ij} + \epsilon}{\sum_{k \in \mathcal{I}} (b_{kj} + \epsilon)}$$

Matrix Factorization

Utilizan un modelo estático de matrix factorization donde construyen una matriz sparse a partir del logaritmo del numero de entradas totales de cada usuario a cada ítem. Este modelo es colaborativo pero ignora la naturaleza secuencial del dataset.

Hidden Markov Model (HMM)

A diferencia de una cadena de Markov, donde conozco el estado del sistema, en una HMM no conozco los estados del sistema, pero si la salida. Esta salida depende de los estados y de sus distribuciones de probabilidad, por lo que usando el método de máxima verosimilitud se puede el modelo.

Brightkite

Red social basada en lugares donde los usuarios podían hacer "check-in" de donde estaban para encontrar amigos. Los logs tienen la forma: (user-id, location-id, check-in time)

LastFM

Secuencias de canciones escuchadas por usuarios. En este trabajo utilizaron las secuencias por artista.

Ambos dataset se preprocesan, quedándose solo con usuarios que tienen al menos 10 observaciones e items que se consumieron al menos 10 veces.

También se eliminan los usuarios que consumieron solo un item muchas veces. El test se obtuvo a partir 2.5% final de las secuencias.

En el caso de LastFM, se dejó como largo máximo de las secuencias 5120 observaciones.

La forma final de los datasets es la siguiente:

	Users	Items	Events	$E_{Unif.}$
BK	1 679	23 243	599 618	10.05
LFM	954	48 188	4 320 170	10.78

Donde E_{Unif} es la probabilidad log negativa asignada a las secuencias por el baseline uniforme.

Gated recurrent unit (GRU) es un tipo de RNN que agrega dos vectores (update y reset) que actuan como compuerta para solucionar el problema del desvanecimiento del gradiente, por lo que el autor decide utilizar una RNN GRU como para su modelo en vez de una RNN basica (tanh RNN).

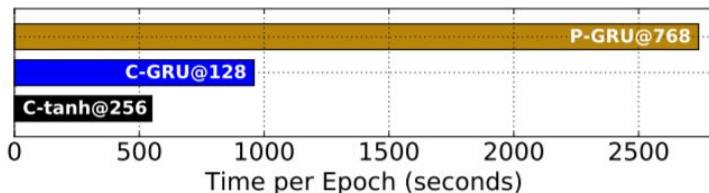
El autor además compara su Collaborative GRU (C-GRU) con los siguientes modelos:

- GRU usuario-agnostica (P-GRU)
- RNN basica colaborativa (C-tanh)

Comparacion entre RNN

Los resultados entre las distintas RNN son los siguientes:

	P-GRU	C-tanh	C-GRU
BK	6.45	6.08	6.02
LFM	4.58	4.90	4.51



Comparacion entre las baselines

Los resultados entre las distintas baselines son los siguientes:

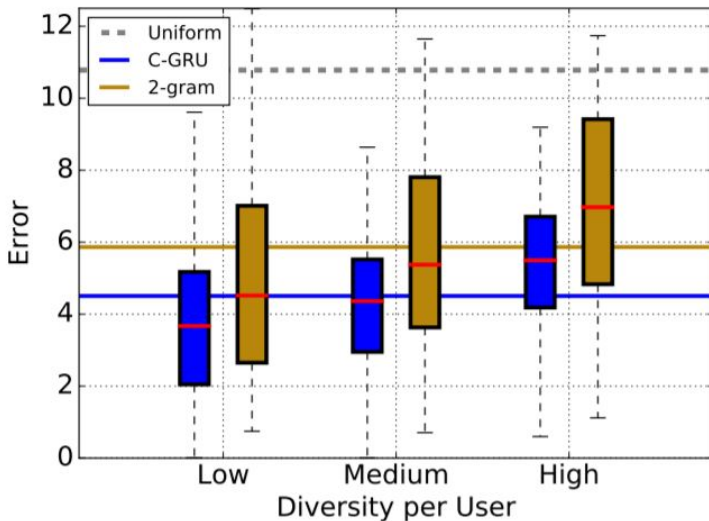
	1-gram	MF	HMM	2-gram	C-GRU
BK	9.53	9.40	8.81	6.73	6.02
LFM	8.60	8.86	7.66	5.87	4.51

Error en relación a la dificultad de la secuencia

Se separaron los usuarios en 3 bins de acuerdo a su entropía de Shannon calculada de acuerdo a la distribución de items que el usuario consume.

Caracterización del error

El resultado es el siguiente:



- La collaborative RNN fue capaz de analizar secuencias y obtener mejores resultados que el baseline en dos datasets distintos.
- El modelo es practico ya que puede ser escalado a datasets grandes aprovechando paralelización

- Incorporar al modelo el tiempo absoluto para ampliar la contextualización.



Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. “Collaborative Recurrent Neural Networks for Dynamic Recommender Systems”. In: *Proceedings of The 8th Asian Conference on Machine Learning*. Ed. by Robert J. Durrant and Kee-Eung Kim. Vol. 63. Proceedings of Machine Learning Research. The University of Waikato, Hamilton, New Zealand: PMLR, 16–18 Nov 2016, pp. 366–381. URL: <http://proceedings.mlr.press/v63/ko101.html>.