

Adapting to Natural Rating Acquisition with Active Learning Strategies (2012)

Mehdi Elahi, Francesco Ricci, and Neil Rubens

Presentador: Javier Machin

Enfocan su trabajo usando Filtrado-Colaborativo

- Recomendar ítems a otros usuarios basándose en los rating que tiene el usuario en común con otros usuarios.

Problema

- La precisión de un Sistema Recomendador (SR) depende de la cantidad de rating y la calidad de los rating.
- Los usuarios frecuentemente clasifica ítems que no son los mejores.

Usar técnicas de Active Learning (AL) para mejorar la calidad de los rating seleccionando

- Solicitando los ratings más informativos.

Trabajos previos

- Métodos de AL no personalizados:
 - Seleccionar el ítem mas popular ó simplemente el que mayor diversidad de ratings tenga (i.e, alta entropía)
- Métodos de AL personalizados:
 - Árboles de decisión para seleccionar el ítem que se le recomendará al usuario basándose en la información previa de los rating del usuario.
 - Enfocados principalmente en el problema del Cold-Start.

El problema de Cold-Start no es tan terrible.

- El usuario empieza a interactuar con el sistema y comienza a evolucionar.
- El verdadero problema surge a la hora de encontrar cual es el mejor ítem que se le ofrecerá al usuario en cada iteración subsiguiente al Cold Start .
 - Utilizar técnicas de AL.

Combinan un AL con la adquisición natural de los ítems y lo evalúan a lo largo del SR

Solo una técnica de AL no es suficiente para mejorar el rendimiento

- Proponen un híbrido.
 - Voting + la mejor estrategia de AL:

Estrategias

- $\text{Log}(\text{popularidad}) * \text{Entropía}$
- Highest Predicted
- Random
- Combined with Voting

Log(popularidad)*Entropía

- i.e, el puntaje para el ítem (i) se calcula multiplicando el logaritmo de la popularidad (el número de rating de i) por la entropía (Nos permite generar un número que nos indica el grado de desorden) del rating (i).

Log(popularidad)*Entropía

- Este método combina el efecto de la popularidad con la heurística que favorece a los ítems con mayor diversidad entre los ratings que provee información acerca de las preferencias del usuario.

Highest Predicted

- Utiliza los rating en el set de train.
- Las predicciones son calculadas para todos los ítems del Usuario.
- Utiliza una factorización matricial con gradientes descendentes optimizados para generar las predicciones de los ratings.

Random

- El puntaje de los ítems es un entero aleatorio
- Esta estrategia es usada para la comparación.

Combined with Voting

- El score del ítem (i) es el número de votos dado por el conjunto de las estrategias antes mencionadas
 - Ejemplo:
 - Cada estrategia obtiene 100 candidatos, entonces los ítems más seleccionados en la lista son los utilizados.

Evaluación:

- Diseñan un procedimiento para simular la evolución del rendimiento del SR
 - Mezclando los ratings adquiridos por las estrategias del AL con los ratings agregados de forma natural por los usuarios que no han sido consultados.
- Utilizan una versión del data set de MovieLens de 1 millón de rating.
- Solo consideran los usuarios activos de los últimos 2 meses.

Evaluación:

- Dividieron los datos en 3 matrices: K, X, T:
 - K: es el set de rating conocidos, adquiridos por MovieLens en el mes 1 (14.195 ratings)
 - El resto es dividido de manera random:
 - 70 % en X (251.241)
 - 30 % en T (111.867)

Evaluación:

- X son los rating que están disponibles para ser obtenidos
- T son los rating que nunca fueron obtenidos y son usados para probar el rendimiento del sistema.

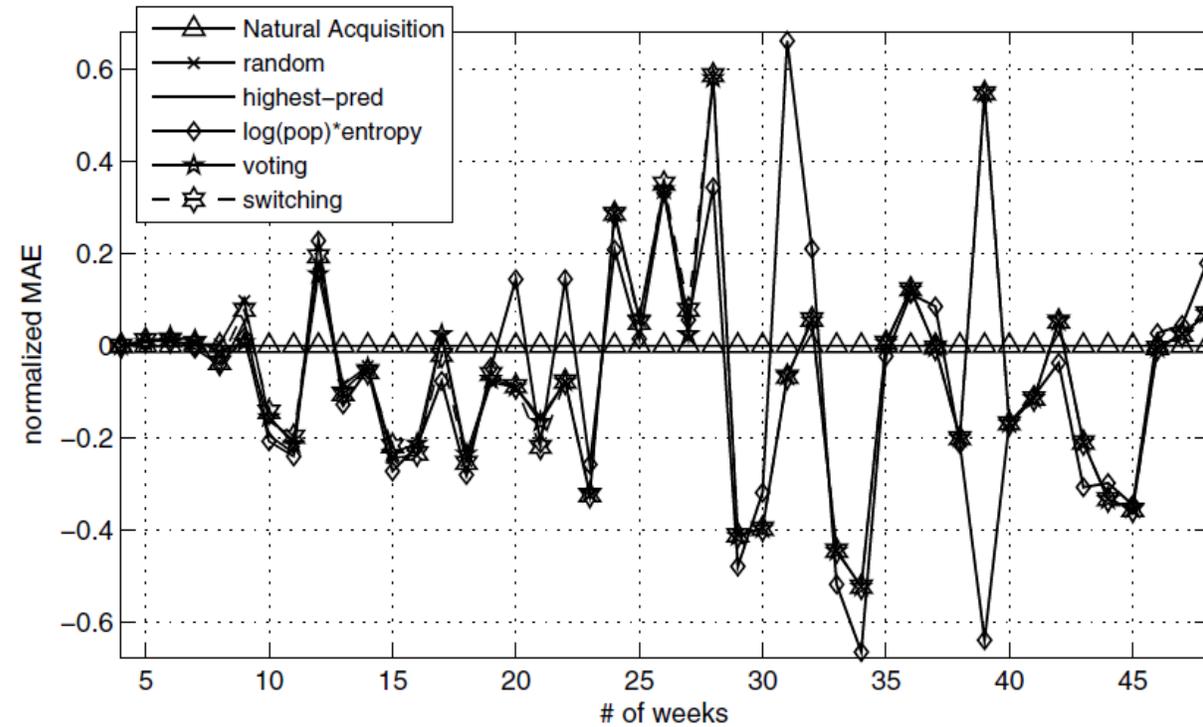
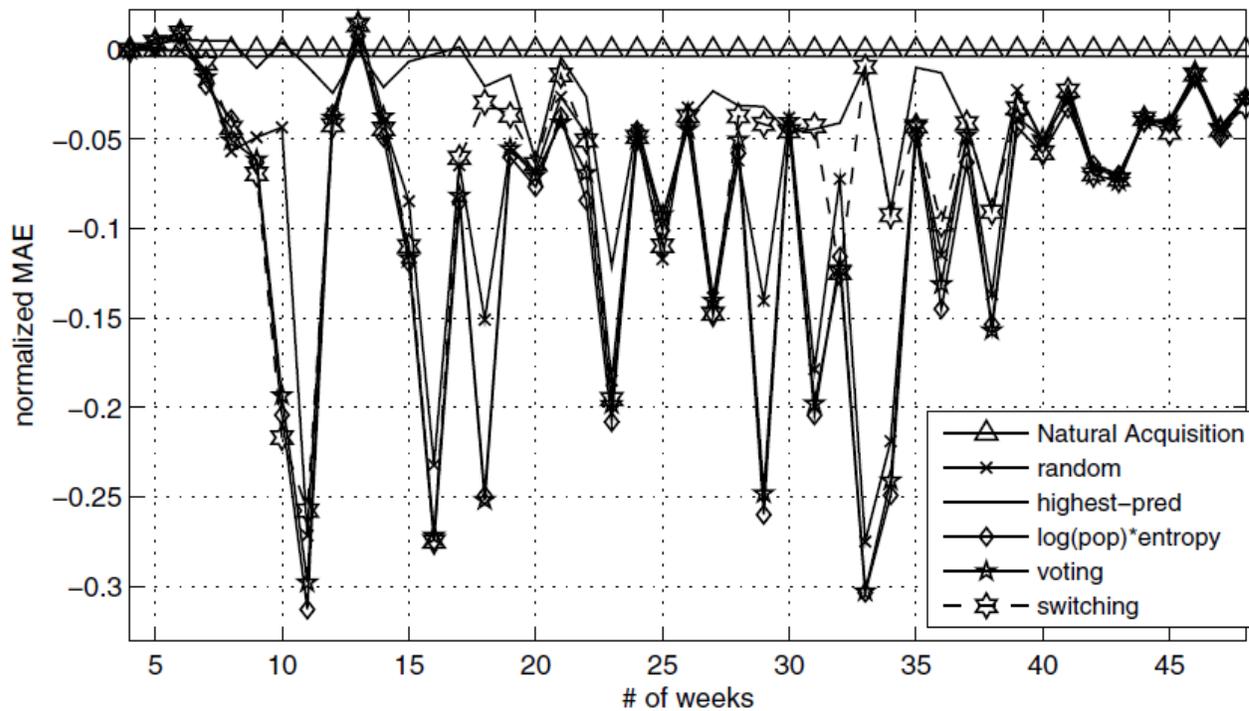
Evaluación:

- Cada día de la semana le piden al usuario que clasifique 40 ítem:
 - El usuario debe tener ítems en K ya clasificados.
- Si los ranking están presentes en X entonces son adicionados a K
- Este proceso lo repiten por 7 días

Evaluación:

- Después se agregan todos los rating en el data set de MovieLens, que se adquirieron en la semana.
 - i.e, son agregadas a K, los rating agregados por los usuarios.
- Finalmente el sistema se entrena usando los rating de K y realiza las pruebas en los rating de T, que el usuario clasifico en la semana.
- Este proceso lo repiten por un año.

Evolución del MAE en las primeras 48 Semanas



- Rating agregados por las estrategias

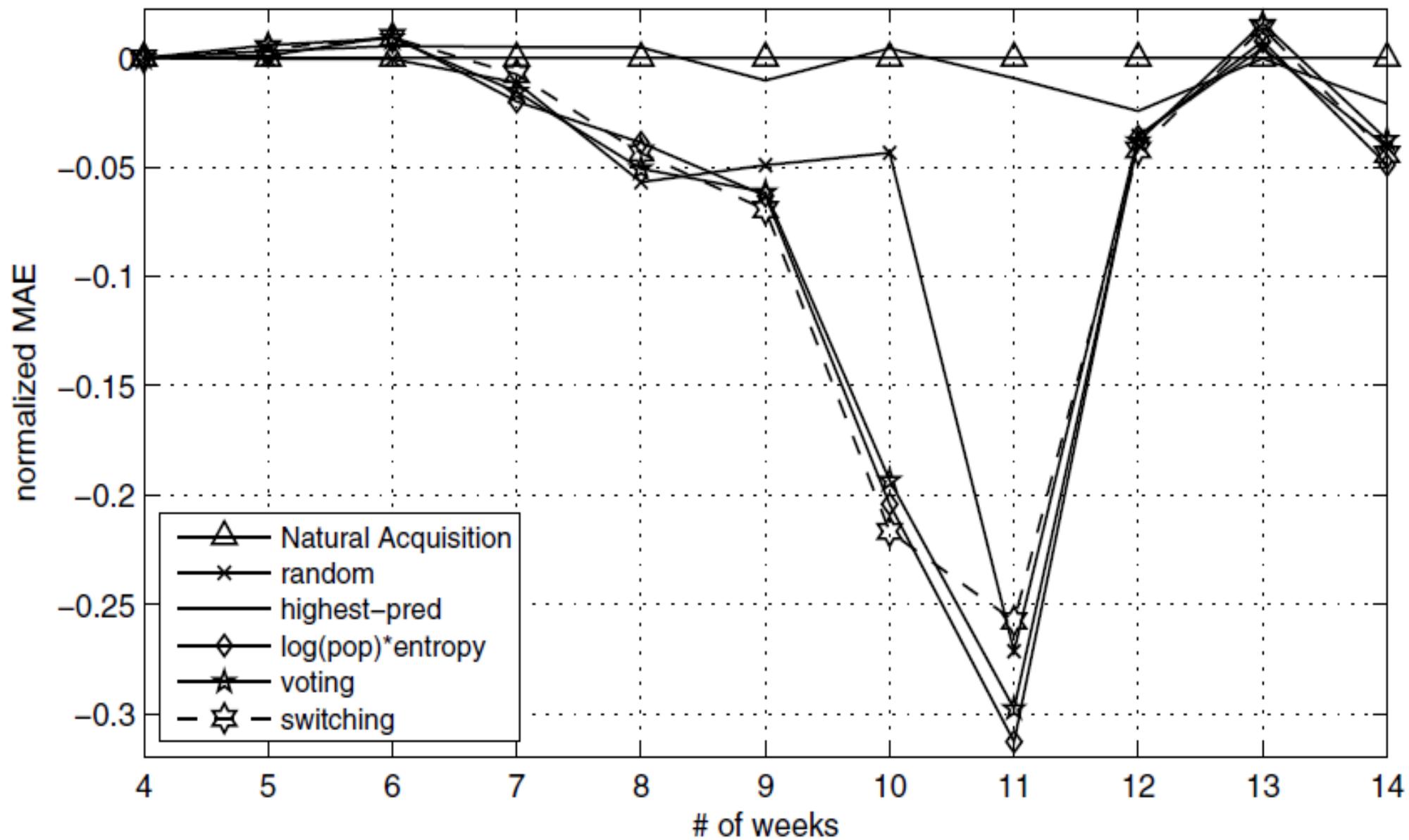


Fig. 3. Normalized system MAE under the effect of AL strategies and natural acquisition (14 weeks)

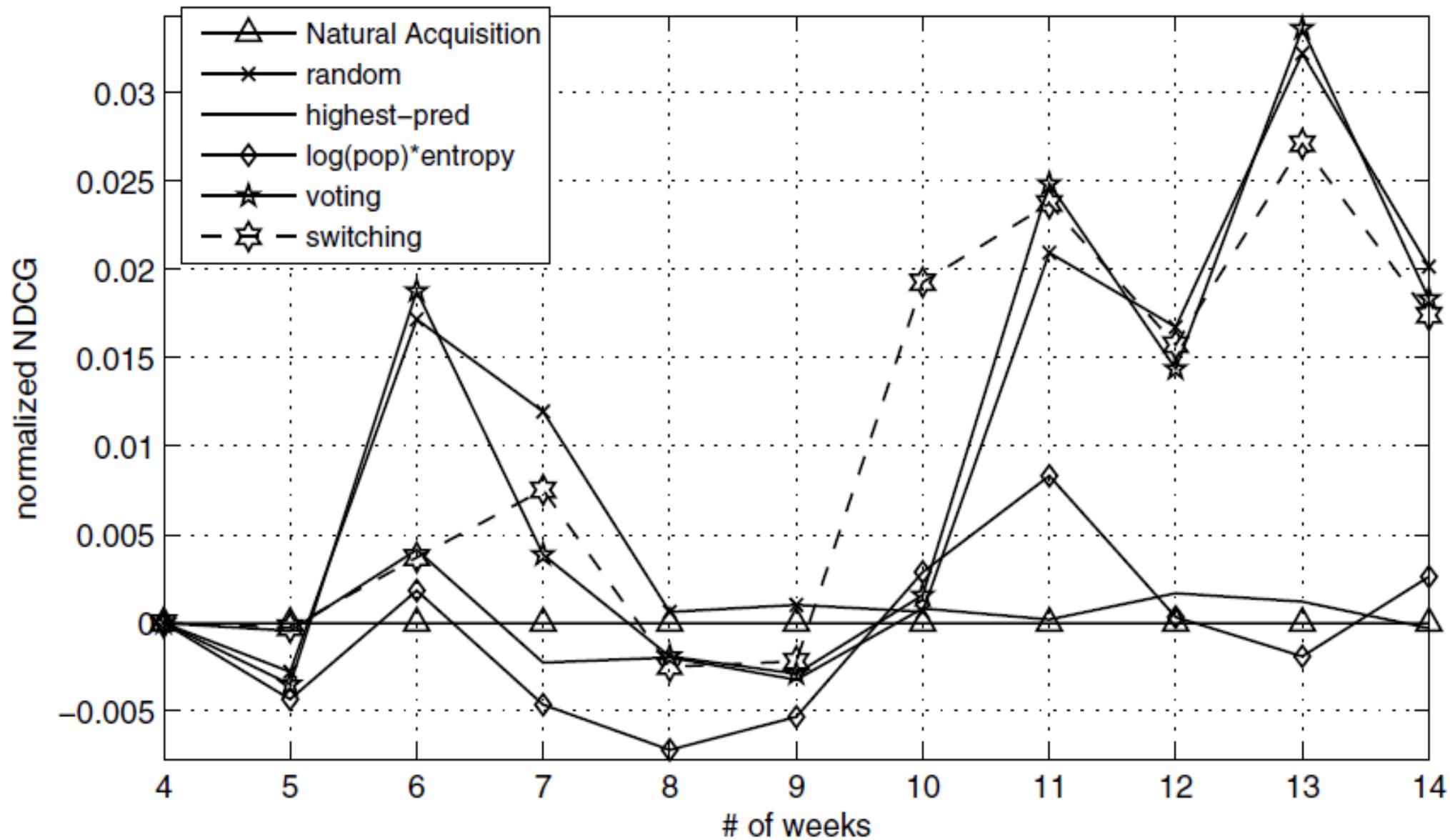


Fig. 4. Normalized system NDCG under the effect of AL strategies and natural acquisition (14 weeks)

Número de semanas en que cada estrategia es la mejor ó la segunda mejor

Table 1. Number of weeks that a strategy performs as the best or second best (in 48 weeks)

	Without Nat. Acquisition		With Nat. Acquisition	
Strategy	MAE	NDCG	MAE	NDCG
Random	34	42	13	16
Highest predicted	20	28	8	10
log(popularity)*entropy	22	28	29	20
Voting	26	42	18	23
Switching (40% exploration)	22	36	15	20

Active Learning Aspect model in Recommender System (2011)

Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos, Lars
Schmidt-Thieme

Presentador: Javier Machin

Nota

- Según lo enseñado al inicio de la presentación:
 - ¿Cual es la diferencia entre el Clásico Active Learning, y el Active Learning que se hace con SR?

Método de AL desarrollado especialmente basado en el modelo de aspecto de características

Problema

- El rendimiento de un SR depende del número de rating que los usuarios provean:
 - Existe casos donde los usuarios no están disponibles a ofrecer rating para una gran cantidad de ítems.
 - Problema del Cold-Start.
- Las consultas que le son realizadas al usuario no son cuidadosamente seleccionadas.

Proponen una técnica de AL para adquirir esos ratings y obtener el interés del usuario.

- Compiten con un enfoque bayesiano en cuanto a Precisión, y evalúan que sucede cuando los resultados son dramáticamente reducidos.

Motivación

- El AL en SR permite mejorar el rendimiento y la precisión.
- El tiempo de respuesta será mucho menor:
 - Especialmente con los usuarios nuevos que no se sabe que preguntarle

Usan un AL basado en el Modelo de Aspecto.

- El modelo de aspecto tiene características especiales que necesitan su propio criterio de AL.
- Los AL en SR deben ser rápidos y precisos:
 - Existen AL basados en la entropía que tiene ventajas de ser preciso pero lamentablemente son muy costosos y se demoran mucho tiempo.

Exploran alternativas utilizando la precisión de los Model_Base en SR

- Model_Base:
 - Usa patrones de rating de los usuarios en el set de entrenamiento.
 - Model_Base trabaja mejor con los datos sparse, que Memory_Base y a la ves es mas escalable.

Contribuciones

- Proponen un criterio de selección que esta basado en un modelo Aspect_Feature con AL
- Reducen el tiempo de consulta:
 - i.e, el algoritmo toma en cuenta la diferencia entre el clásico AL y el AL para SR. 😊

Model_Aspect

- Es un modelo probabilístico de espacios latentes que modela preferencias individuales como la combinación convexa de un factor de preferencias.
 - $z \in Z = \{z_1, z_2, z_3, \dots, z_k\}$ asociado con cada par de (usuario) e (ítem)

Model_Aspect

- El modelo aspecto asume que los usuarios y los ítems son independientes.
 - La probabilidad de (m, u, r) es calculada como:

$$p(r|m, u) = \sum_{z \in Z} p(r|z, m)p(z|u)$$

- Los rating de los usuarios son normalizados, serán una distribución normal con media 0 y varianza 1.

Bayesian Active Learning

- Aunque el sistema tenga un modelo global preciso, el modelo del nuevo usuario es débil.
 - El sistema tiene muy pocos rating disponibles

Bayesian Active Learning

- Para seleccionar el ítem con AL ellos analizan las minimizaciones de entropía.

Bayesian Active Learning

$$m_u^* = \operatorname{argmin}_{m \in M} - \left\langle \sum_{z \in Z} \theta_{u_z | m, r} \log \theta_{u_z | m, r} \right\rangle_{P(r|u, m)}$$

- θ_{Uz} : denota los parámetros actuales del usuario ($P(Z|U)$)
- $\theta_{Uz|m,r}$: denota los parámetros después de entrenar el modelo_user basado en el rating (r) obtenido a partir de la película (m) del usuario, i.e, $P(z|u,m,r)$

Bayesian Active Learning

$$m_u^* = \operatorname{argmin}_{m \in M} - \left\langle \sum_{z \in Z} \theta_{u_z | m, r} \log \theta_{u_z | m, r} \right\rangle_{P(r|u, m)}$$

- Intenta categorizar los usuarios en una sola clase, mientras que en la realidad puede estar en varias clases. ☹
- Ejemplo: La película puede ser cómica o de acción.

Supuestos de AL propuesto

- Considerar el método de predicción de rating en el "Aspect_Model" y entonces optimizarlo.
- Los rating no conocidos en el "Aspect_Model" son estimados con:

$$p(r|m, u) = \sum_{z \in Z} p(r|z, m)p(z|u)$$

Supuestos de AL propuesto

- Después de obtener un nuevo rating, los parámetros son reentrenados.
 - i.e, al actualizar los parámetros se obtiene mejores rating no conocidos.

Supuestos de AL propuesto

- Asumen que los ítem siguen una distribución normal.

$$\mu_{m,z} = \frac{\sum_{\langle u, m', r \rangle: m'=m} r P(z|u, r, m)}{\sum_{\langle u, m', r \rangle: m'=m} P(z|u, r, m)}$$

$$\sigma_{m,z}^2 = \frac{\sum_{\langle u, m', r \rangle: m'=m} (r - \mu_{m,z})^2 P(z|u, r, m)}{\sum_{\langle u, m', r \rangle: m'=m} P(z|u, r, m)}$$

Supuestos de AL propuesto

- En promedio cada ítem tiene ratings suficientes entre los usuarios de entrenamiento.
 - i.e, los artículos han entrenado lo suficiente en la fase de formación y actualización de parámetros de los ítems.
- Un nuevo rating del usuario no afecta.

Supuestos de AL propuesto

- El nuevo rating es muy relevante para los nuevos parámetros de usuario.
 - Estos parámetros ya han sido capacitados con algunas calificaciones y su actualización después de cada rating nuevo provoca una gran cantidad de mejoras.

Supuestos de AL propuesto

- Los parámetros son obtenidos :

$$P(z|u, r, m; \hat{\theta}) = \frac{\hat{p}(r|m, z) \hat{P}(z|u)}{\sum_{z'} \hat{p}(r|m, z) \hat{P}(z'|u)}$$

- Los parámetros latentes de usuario (u) están entrenados con todas las tuplas (u, m, r) relevantes para (u) y los calculan como:

$$P(z|u) \propto \left(\mu_{m,z}, \sigma_{m,z}^2 \right)_{\forall m \in M}$$

Supuestos de AL propuesto

- La mejora que se obtiene después de rentrenar los parámetros de usuario, depende del elemento que se utiliza para el rentrenar.
- Mayor certeza sobre media y la varianza del elemento conduce a una mayor mejora en los parámetros de usuario.

Supuestos de AL propuesto

- Nota: De estadística sabemos que la certeza acerca de media y la varianza depende del número de muestras.
 - i.e, la validez de media y varianza aumenta a medida que el número de muestras crece.
- El artículo más popular tiene más certeza de la media y la varianza.

Experimentos

- Movielense: 943 usuarios, 1682 ítems
- MovieRating: 500 usuarios, 1000 ítems

- Los dataset fueron divididos de manera random entre los set de train y test

- Train:
 - Movielense: 343 usuarios
 - MovieRating: 200 usuarios

Experimentos

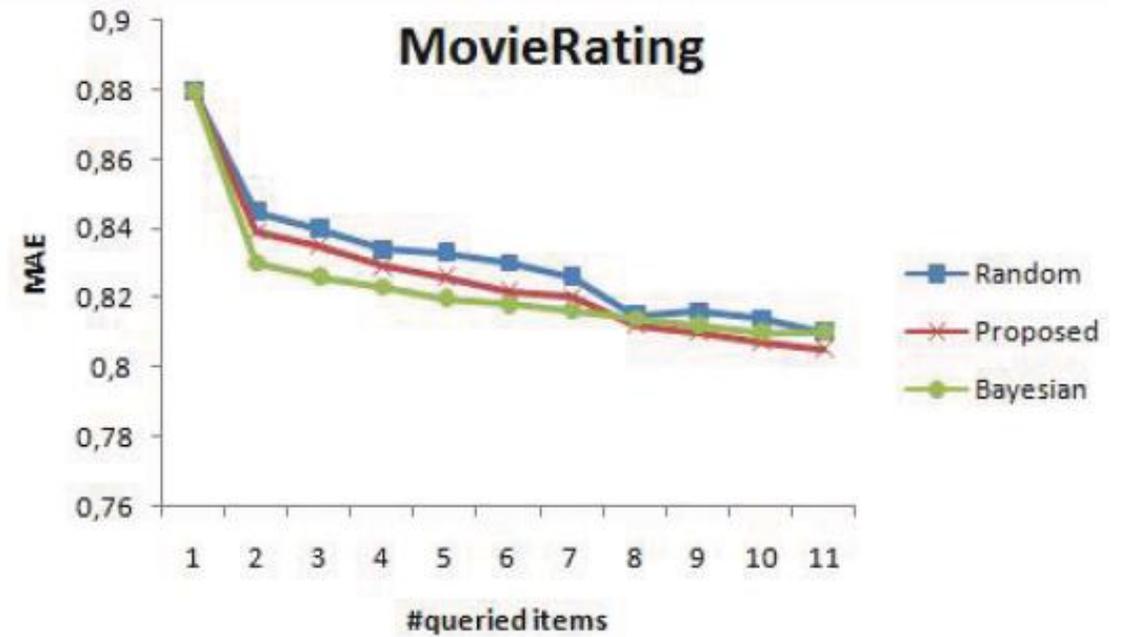
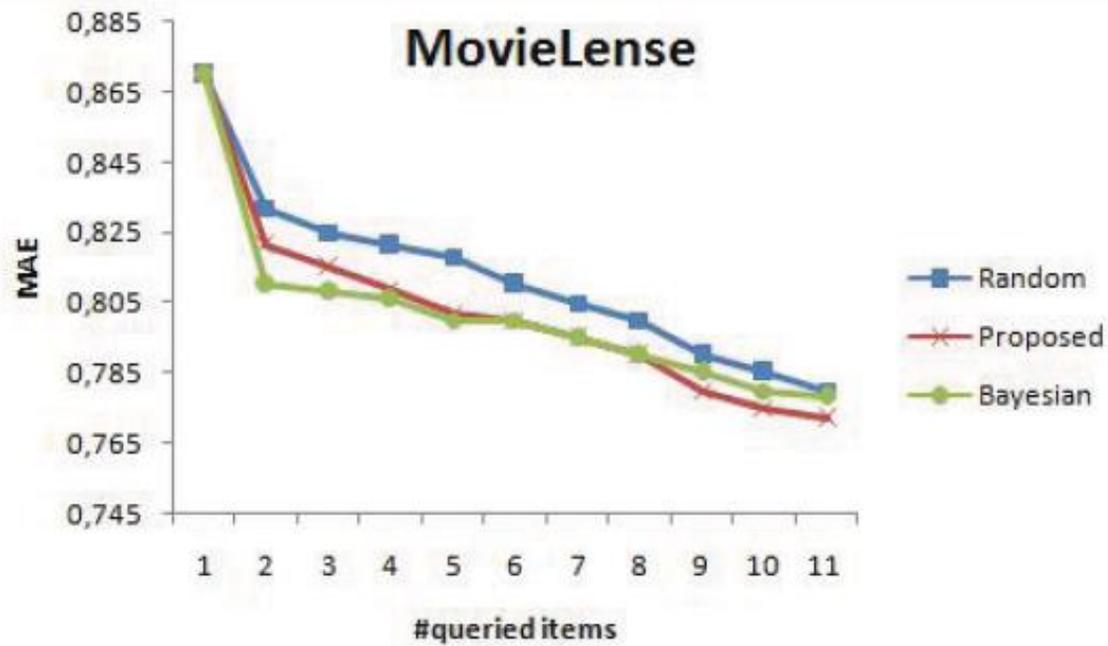
- Test:
 - Utilizaron usuario que hayan clasificado más de 33 ítems, de los cuales al menos 20 de ellos son usados como datos de test y que a la vez ya se encuentran en los datos de entrenamiento.
- Cada usuario de test es considerado como nuevo, el modelo preliminar de cada uno es construido usando 3 clasificaciones iniciales random.
 - El algoritmo de AL selecciona ítems del pool.

Calculan el Accuracy usando MAE (Mean Absolute Error)

$$MAE = \frac{1}{N_{Test}} \sum_{m \in M_{Test}} |r^{true} - r^{predicted}|$$

- N_{test} : es el número de ítems de test
- M_{test} : es el set de ítem de test

Resultados

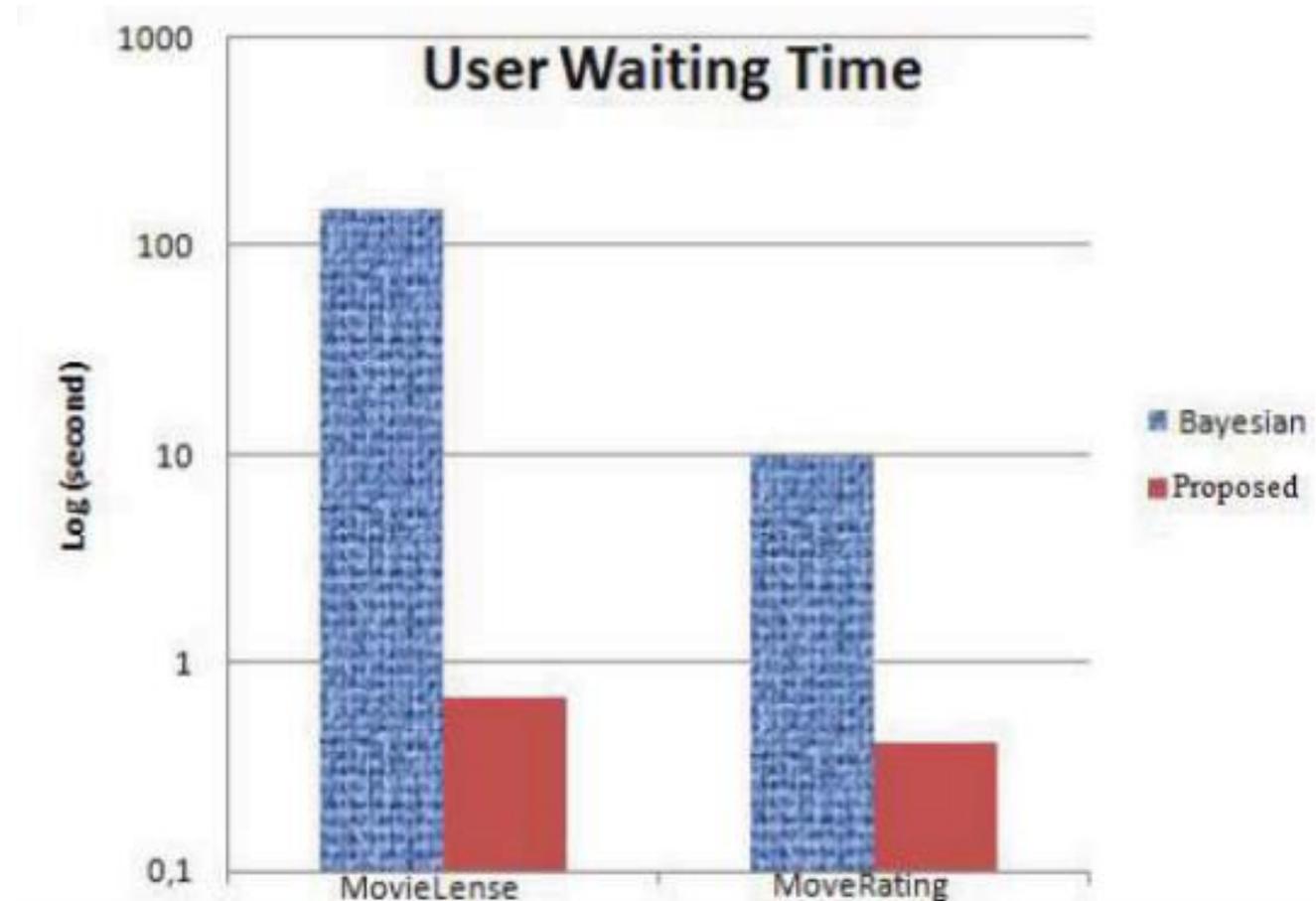


Resultados

- La primera parte: El método **bayesiano** es mejor. 😊
 - Ocupa de la atracción del cliente, tiene como objetivo encontrar nuevos clientes.
- La segunda parte: El método **propuesto** es mejor. 😊
 - Satisfacción del cliente, se encarga de mantener a los clientes actuales.

Resultados

- Tiempo en método propuesto es mucho menor que el método Bayesiano.
- Método bayesiano necesita una gran cantidad de cálculo para la estimación la posterior 😞.



A Contextual-Bandit Approach to Personalized News Article Recommendation (2010)

Lihong Li, Wei Chu, John Langford, Robert E. Schapire

Presentador: Javier Machin

Los servicios web personalizados se esfuerzan por adaptar sus servicios (anuncios, artículos de prensa, etc.) a usuarios individuales, haciendo uso de los contenidos y la información del usuario.

Problemas

- El servicio web se ofrece con cambio dinámico en un pool de contenido, lo que provoca que los métodos sean intratables.
- Exige soluciones que sean a la vez rápido en el aprendizaje.

Motivación

- Identificar rápidamente el contenido más interesante en la web para los usuarios.

Propuesta

- Modelan la recomendación personalizada de un nuevo artículos de noticias como un problema contextual "bandit",

Bandit

- Es un approach en el cual un algoritmo de aprendizaje, secuencialmente selecciona los artículos más relevantes para los usuarios:
 - Usando la información contextual de los usuarios y artículos.
 - Al mismo tiempo adaptar su estrategia de selección de artículo basado la retroalimentación de usuario.
 - en este caso, usan el "click " del mouse.

Bandit

- Personalizar un nuevo artículo para recomendar, puede ser modelado como un problema de “multi armed bandit” con información contextual.
- $T = \text{Ensayos ó Tiradas}$
 - $T = 1, 2, 3, \dots$

Bandit

- El algoritmo observa al usuario actual (U_t) y a un set (A_t) de arms (brazos) ó acciones junto con el vector de características ($X_{t,a}$).
 - El vector ($X_{t,a}$) resume la información del usuario (U_t), y la acción del brazo (a) será referenciado como el contexto.
- Utiliza los beneficios observados en tiradas anteriores:
 - (A) Selecciona un brazo ($a_t \in A_t$) y recibe las ganancias de (R_{t,a_t}), cuya expectativa depende del usuario (U_t) y del brazo (a_t).

Bandit

- El algoritmo entonces realiza la selección estratégica del brazo con la nueva información.

$$(X_t, a_t, Q_t, R_t, a_t)$$

- Nota: No existe retroalimentación para los brazos no elegidos

Algoritmo LinUCB

Algorithm 1 LinUCB with disjoint linear models.

```
0: Inputs:  $\alpha \in \mathbb{R}_+$ 
1: for  $t = 1, 2, 3, \dots, T$  do
2:   Observe features of all arms  $a \in \mathcal{A}_t$ :  $\mathbf{x}_{t,a} \in \mathbb{R}^d$ 
3:   for all  $a \in \mathcal{A}_t$  do
4:     if  $a$  is new then
5:        $\mathbf{A}_a \leftarrow \mathbf{I}_d$  ( $d$ -dimensional identity matrix)
6:        $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$  ( $d$ -dimensional zero vector)
7:     end if
8:      $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a$ 
9:      $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$ 
10:   end for
11:   Choose arm  $a_t = \arg \max_{a \in \mathcal{A}_t} p_{t,a}$  with ties broken arbitrarily, and observe a real-valued payoff  $r_t$ 
12:    $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t} \mathbf{x}_{t,a_t}^\top$ 
13:    $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$ 
14: end for
```

- UCB : Upper Confidence Bound
- \mathcal{A}_t : set de brazos
- b_a : número de click/no click
- $\mathbf{x}_{t,a}$: set de características en la tirada (t) en el brazo (a)

$$\hat{\boldsymbol{\theta}}_a = (\mathbf{D}_a^\top \mathbf{D}_a + \mathbf{I}_d)^{-1} \mathbf{D}_a^\top \mathbf{c}_a,$$

