Bayesian Personalized Ranking (BPR)

Denis Parra

PUC Chile

IIC3633

BPR: Bayesian Personalized Ranking from Implicit Feedback

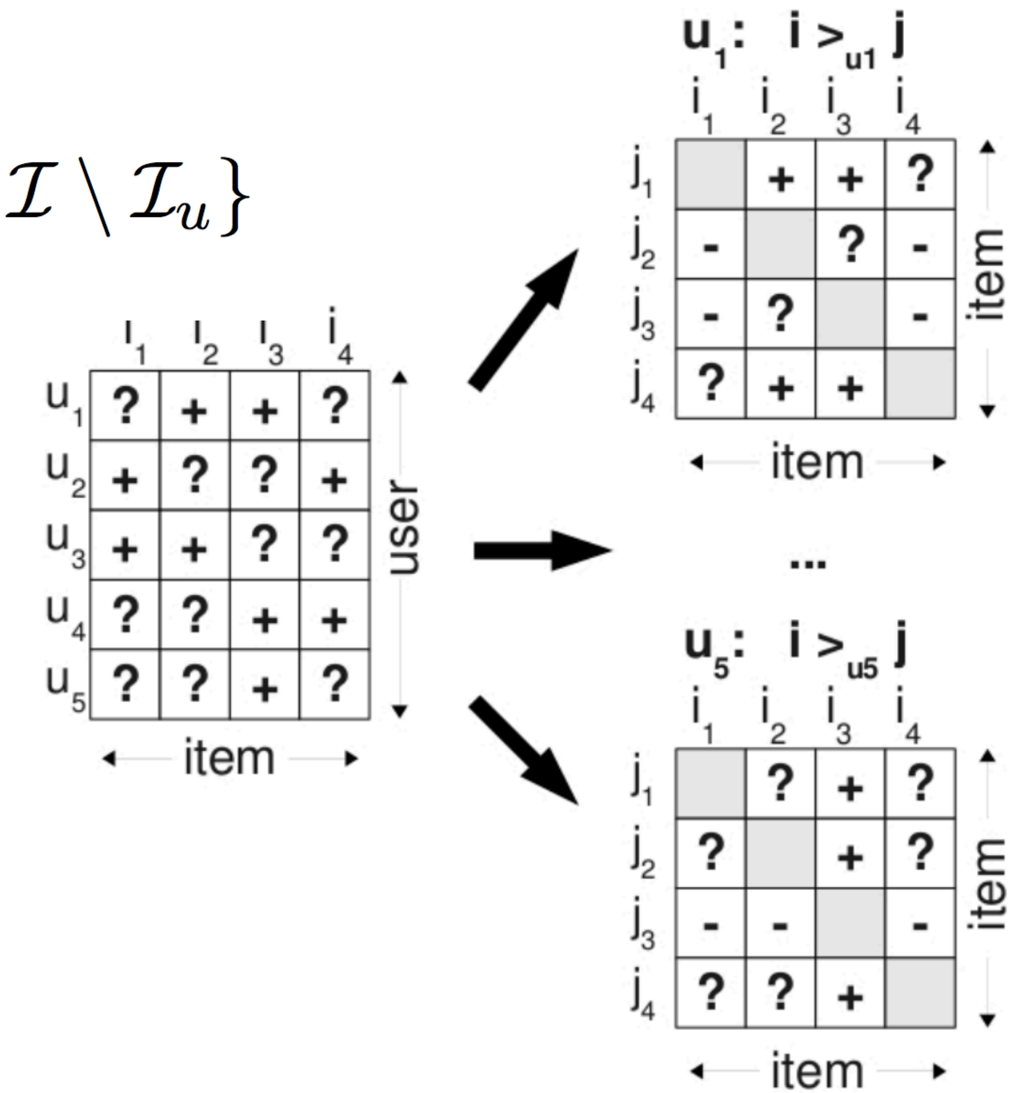
Steffen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme
{srendle, freudenthaler, gantner, schmidt-thieme}@ismll.de
Machine Learning Lab, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany

Introducción

- Hasta el momento hemos visto el problema de recomendación como una predicción de score (o rating) que un usuario dará a un ítem. Ejm: score(u,i) vs. score(u,j)
- Sin embargo, es natural pensar en el problema más bien como ordenamiento: Dado un usuario y una lista de ítems, el usuario podría ordenarlos según su preferencia, en lugar de indicar el nivel exacto de preferencia por cada ítem. Ejm: score(u,i,j)
- Una solución a este problema es la de aprender directamente una función de ranking personalizada.

Escenario: Feedback sólo positivo

$$\mathcal{D}_p = \{(u,i,j) \in \mathcal{D} | i \in \mathcal{I}_u \land j \in \mathcal{I} \setminus \mathcal{I}_u\}$$



BPR

- Objetivo: Encontrar una función de ranking personalizado.

- Uno de los métodos de “Learning to Rank” más populares.

- BPR por sí mismo no es un algoritmo: más bien una función de pérdida y un framework para llevar a cabo la optimización.

Algoritmo = modelo + función de pérdida + aprensizaje

Ejemplo de algoritmo: BPR-MF

- Modelo: MF (factorización matricial)

- Función de pérdida: BPR-OPT (aproxima AUC)

- Aprendizaje: BPR-Learn (basado en SGD)

Formulación Bayesiana del Problema

- $\succ$: La estructura de preferencias desconocida (ordenamiento)
  - Usaremos el ranking entre pares derivado de los datos D_p
- $>_u$: Preferencias (ranking) del usuario u.
  - Ejm: Si el u prefiere i_2 sobre i_1, luego $i_2 >_u i_1$
- Θ : Parámetros de un modelo de predicción arbitrario
  - En el caso de factorización matricial, $\Theta = W \cup H$

Formulación Bayesiana del Problema

- Bajo la formulación bayesiana, queremos maximizar la siguiente probabilidad "posterior" de Θ , que es el vector de parámetros de un modelo arbitrario:

$$p(\Theta| \succ) \propto p(\succ |\Theta)p(\Theta)$$

- Para el prior, Asumimos comportamiento Gaussiano e independencia de los parámetros

$$\Theta \sim N(0, \tfrac{1}{\lambda}I) \qquad p(\Theta) = \prod_{\theta \in \Theta} \sqrt{\frac{\lambda}{2\pi}}\, e^{-\frac{1}{2}\lambda\theta^2}$$

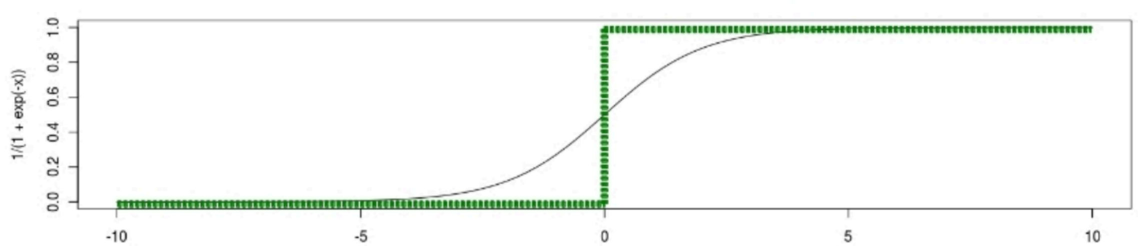
Formulación Bayesiana del Problema II

- Estimación de Máxima Verosimilitud (MLE):
  - Se asume que el feedback de cada usuario es independiente
  - Se asume que cada observación x_{uij} es independiente, luego

$$p(\succ |\Theta) = \prod_{u \in \mathcal{U}} p(\succ_u |\Theta) = \prod_{(u,i,j) \in \mathcal{D}_p} p(i \succ_u j|\Theta)$$

- Usando los scores individuales $\hat{\phi}$

$$p(i \succ_u j|\Theta) = p(\hat{\phi}_{ui} - \hat{\phi}_{uj} > 0) = \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) = \frac{1}{1 + e^{-(\hat{\phi}_{ui} - \hat{\phi}_{uj})}}$$



Estimación MAP

$$
\begin{aligned}
\text{BPR-OPT} &:= \ln p(\Theta | >_u) \\
&= \ln p(>_u | \Theta)\, p(\Theta) \\
&= \ln \prod_{(u,i,j) \in D_S} \sigma(\hat{x}_{uij})\, p(\Theta) \\
&= \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) + \ln p(\Theta) \\
&= \sum_{(u,i,j) \in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2
\end{aligned}
$$

$\hat{x}_{uij}(\Theta)$: una función arbitraria del vector de parámetros que captura la relación entre u, i y j

Si $\hat{x}_{uij} = \hat{\phi}_{ui} - \hat{\phi}_{uj}$, luego:

$$\arg\max_\Theta p(\Theta, \succ) =$$

$$\arg\max_\Theta p(\succ |\Theta) p(\Theta) =$$

$$\arg\max_\Theta ln\, p(\succ |\Theta) p(\Theta) =$$

$$\arg\max_\Theta ln \prod_{(u,i,j) \in \mathcal{D}_p} \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) \sqrt{\frac{\lambda}{2\pi}}\, e^{-\frac{1}{2}\lambda\theta^2}$$

$$\arg\max_\Theta \underbrace{\sum_{(u,i,j) \in \mathcal{D}_p} ln\, \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) - \lambda\|\Theta\|^2}_{BPR\text{–}OPT}$$

SGD

$$
\begin{aligned}
\frac{\partial \text{BPR-O}\textsc{pt}}{\partial \Theta} &= \sum_{(u,i,j)\in D_S} \frac{\partial}{\partial \Theta} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta \frac{\partial}{\partial \Theta} ||\Theta||^2 \\
&\propto \sum_{(u,i,j)\in D_S} \frac{-e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} - \lambda_\Theta \Theta
\end{aligned}
$$

SGD – regla de actualización

$$\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_\Theta \Theta \right)$$

SGD en BPR

<u>LearnBPR</u>

input: f_i, α, Σ^2, *stopping criteria*

initialize $\Theta \sim \mathcal{N}(0, \Sigma^2)$

repeat

  draw $(u, i, j) \in \mathcal{D}_p$ randomly

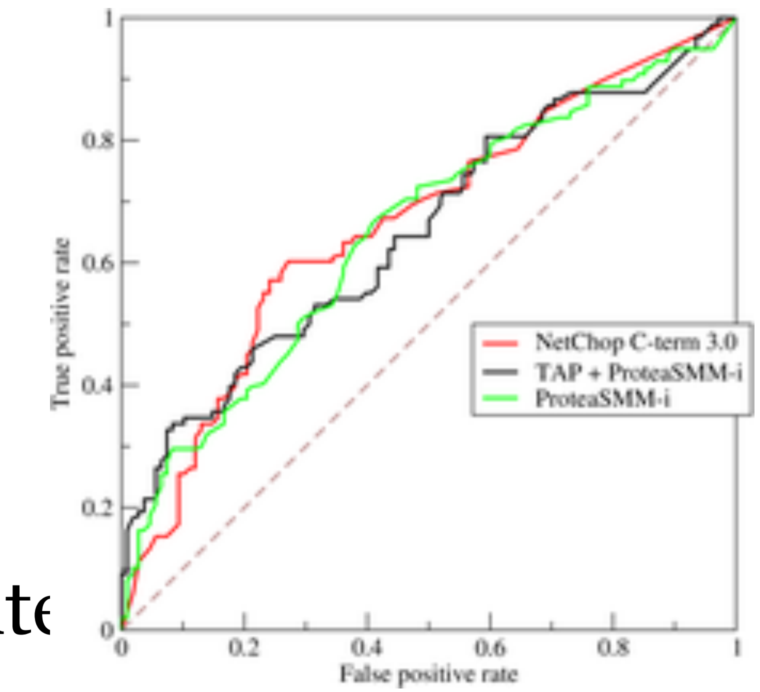
  $\Theta \leftarrow \Theta + \alpha \frac{\partial BPR\text{–}OPT}{\partial \Theta}(\Theta)$

until approximate maximum is reached

return Θ

AUC

- BPR aproxima AUC
- Area Under the Curve: Métrica usada en data mining para calcular la probabilidad de predicciones hechas correctamente
- En BPR: probabilidad de que un par de items muestreados aleatoriamente sean correctamente rankeados (ordenados):



$$AUC = \sum_{u \in \mathcal{U}} AUC(u) = \frac{1}{|\mathcal{U}|} \frac{1}{|\mathcal{I}_u| \; |\mathcal{I} \setminus \mathcal{I}_u|} \sum_{(u,i,j) \in \mathcal{D}_p} \delta(\hat{\phi}_{ui} \succ \hat{\phi}_{uj})$$

$$\delta(\hat{\phi}_{ui} \succ \hat{\phi}_{uj}) = 1 \text{ if } \hat{\phi}_{ui} \succ \hat{\phi}_{uj}, \text{ and } 0, \text{ else}$$

Problema al optimizar AUC

- AUC tiene una forma no diferenciable

- Se acostumbra buscar una función más suave que se aproxime, y usar esa función "proxy" en la optimización

$$\delta(x > 0) = H(x) := \begin{cases} 1, & x > 0 \\ 0, & \text{else} \end{cases}$$

$$\sigma(x) := \frac{1}{1 + e^{-x}}$$

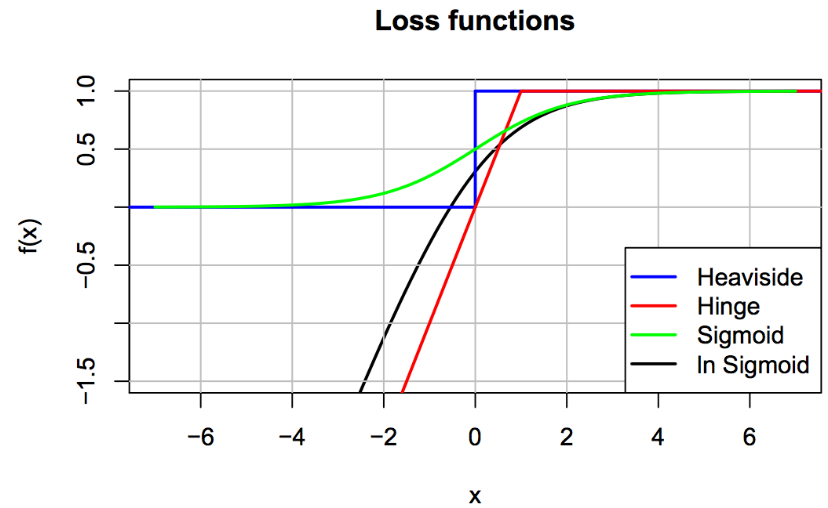


Figure 3: Loss functions for optimizing the AUC. The non-differentiable Heaviside $H(x)$ is often approximated by the sigmoid $\sigma(x)$. Our MLE derivation suggests to use $\ln \sigma(x)$ instead.

Relación entre BPR y AUC

$$AUC - OPT = \sum_{(u,i,j)\in\mathcal{D}_p} \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) - \lambda\|\Theta\|^2$$

$$BPR - OPT = \sum_{(u,i,j)\in\mathcal{D}_p} ln\, \sigma(\hat{\phi}_{ui} - \hat{\phi}_{uj}) \ - \lambda\|\Theta\|^2$$

Algunos tricks en el artículo

- Al ejecutar LearnBPR, no hacer iterar por usuario o por item, sino que elegir tuplas x_{uij} de manera aletoria uniforme.

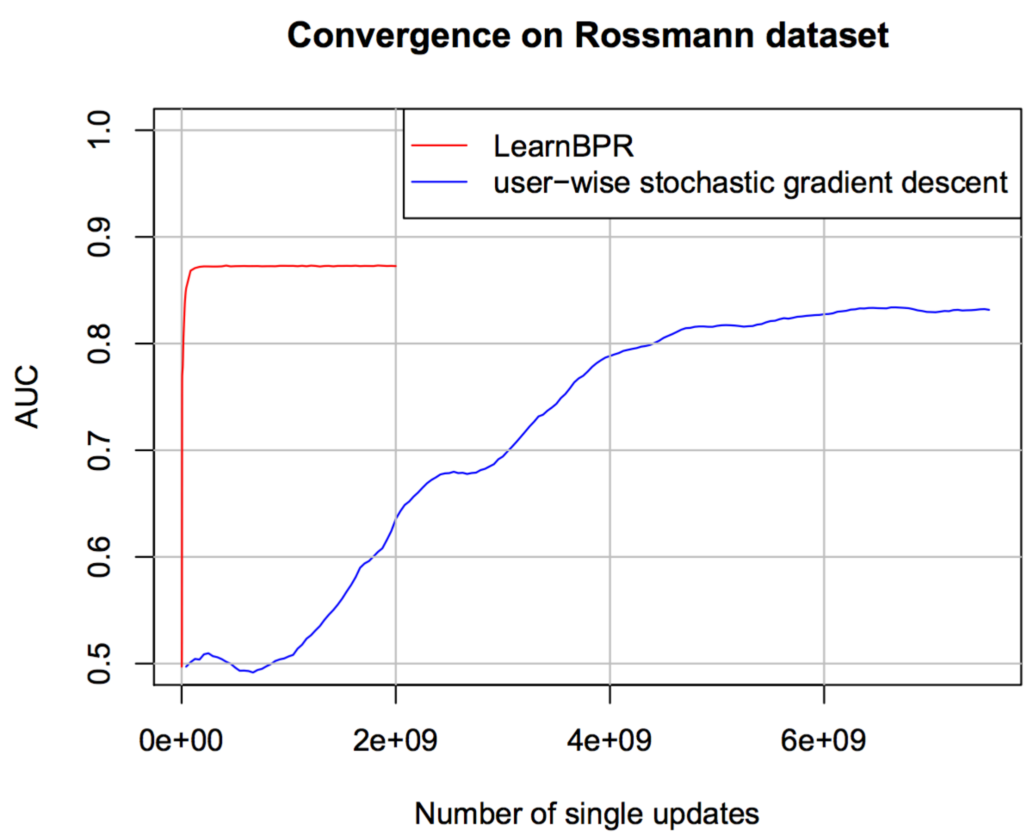
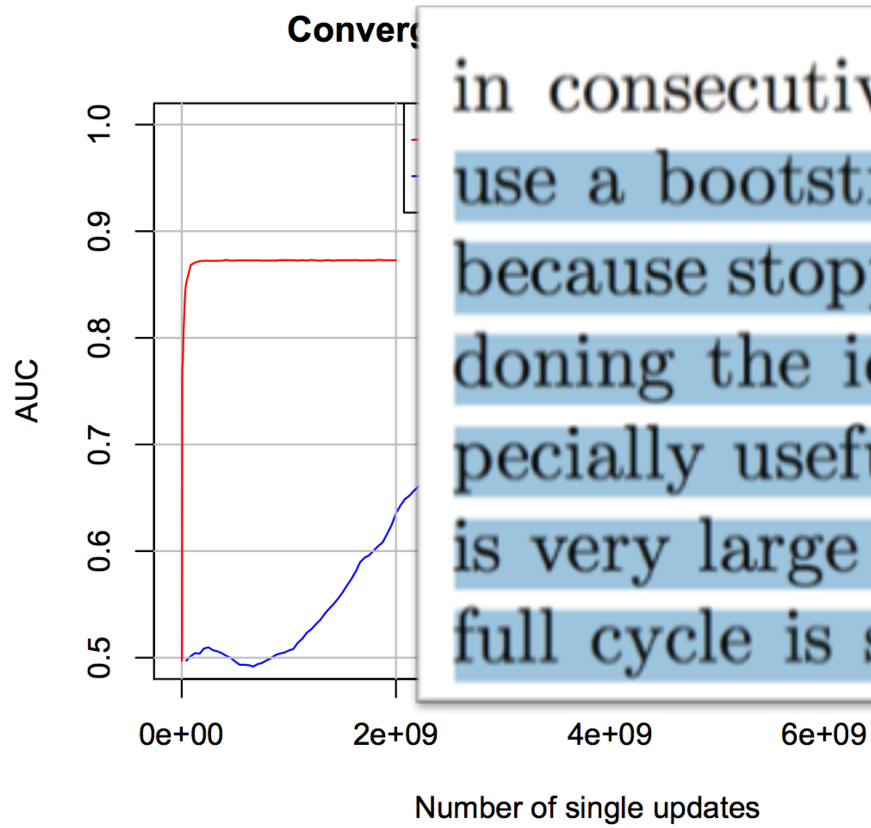
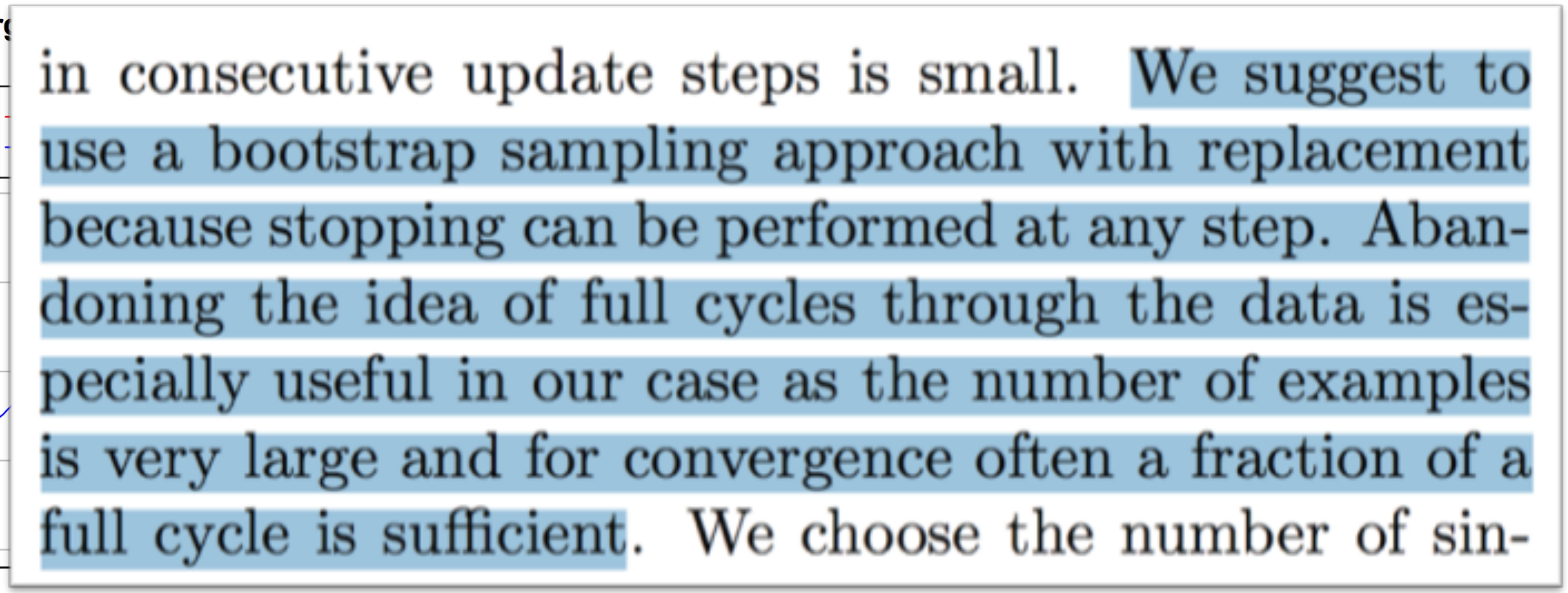


Figure 5: Empirical comparison of the convergence of typical user-wise stochastic gradient descent to our LearnBPR algorithm with bootstrap sampling.

Algunos tricks en el artículo II

- Al ejecutar LearnBPR, no hacer iterar por usuario o por item, sino que elegir tuplas x_{uij} de manera aletoria uniforme.




in consecutive update steps is small. We suggest to use a bootstrap sampling approach with replacement because stopping can be performed at any step. Abandoning the idea of full cycles through the data is especially useful in our case as the number of examples is very large and for convergence often a fraction of a full cycle is sufficient. We choose the number of sin-

Caso BPR-MF

- Definimos $\hat{x}_{uij} := \hat{x}_{ui} - \hat{x}_{uj}$
- Usando Factorización Matricial, tenemos

$$\hat{X} := WH^t \qquad \hat{x}_{ui} = \langle w_u, h_i \rangle = \sum_{f=1}^{k} w_{uf} \cdot h_{if}$$

- Luego, usando BPR-OPT

$$\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{x}_{uij}}}{1 + e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_\Theta \Theta \right) \qquad \frac{\partial}{\partial \theta} \hat{x}_{uij} = \begin{cases} (h_{if} - h_{jf}) & \text{if } \theta = w_{uf}, \\ w_{uf} & \text{if } \theta = h_{if}, \\ -w_{uf} & \text{if } \theta = h_{jf}, \\ 0 & \text{else} \end{cases}$$

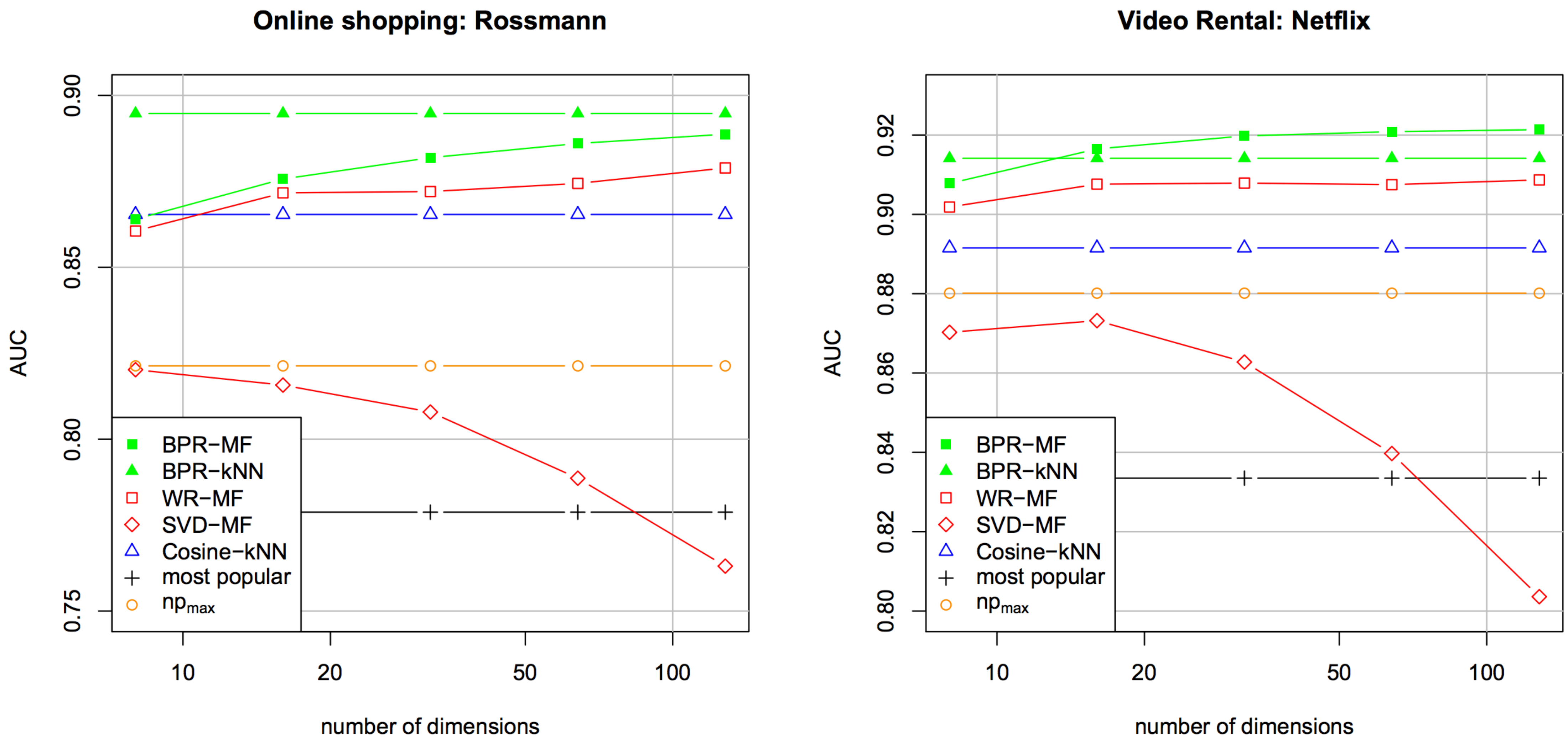
Evaluación

$$\mathrm{AUC} = \frac{1}{|U|} \sum_{u} \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(\hat{x}_{ui} > \hat{x}_{uj}) \qquad (2)$$

where the evaluation pairs per user u are:

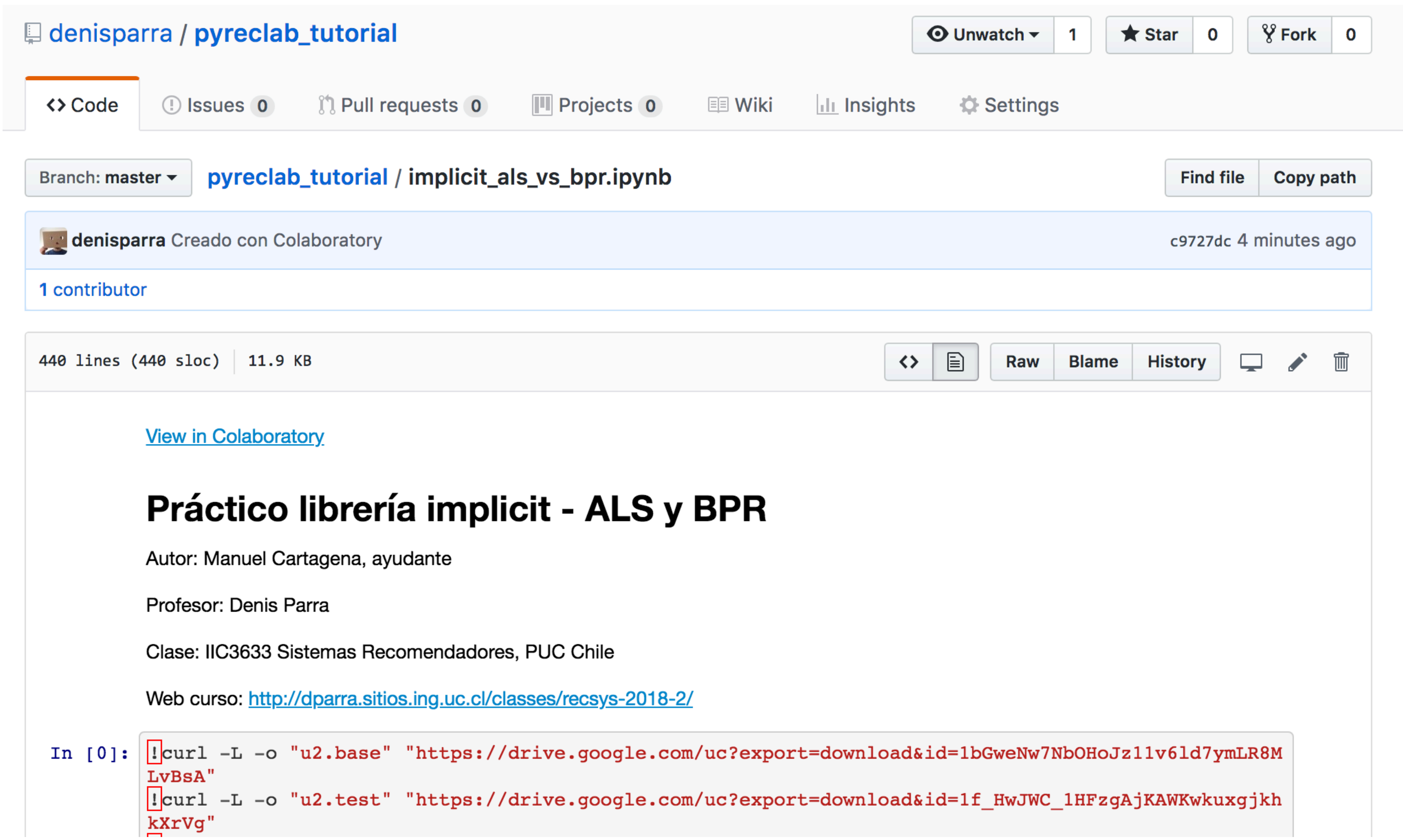
$$E(u) := \{(i,j) | (u,i) \in S_{\mathrm{test}} \wedge (u,j) \not\in (S_{\mathrm{test}} \cup S_{\mathrm{train}})\}$$

Resultados


Online shopping: Rossmann
AUC
0.75
0.80
0.85
0.90
10
20
50
100
number of dimensions
BPR−MF
BPR−kNN
WR−MF
SVD−MF
Cosine−kNN
most popular
np$_{max}$
Video Rental: Netflix
AUC
0.80
0.82
0.84
0.86
0.88
0.90
0.92
10
20
50
100
number of dimensions
BPR−MF
BPR−kNN
WR−MF
SVD−MF
Cosine−kNN
most popular
np$_{max}$

Ejercicio con biblioteca implicit

https://github.com/denisparra/pyreclab_tutorial/blob/master/implicit_als_vs_bpr.ipynb



Gracias!

- dparra@ing.puc.cl

Caso BPR-kNN (Item-based CF)

- Definimos $\hat{x}_{uij} := \hat{x}_{ui} - \hat{x}_{uj}$
- Usando kNN, x_{iu}

$$\hat{x}_{ui} = \sum_{l \in I_u^+ \wedge l \neq i} c_{il}$$, C : I x I matriz simétrica de similaridad

- Luego, usando BPR-OPT

$$\frac{\partial}{\partial \theta} \hat{x}_{uij} = \begin{cases} +1 & \text{if } \theta \in \{c_{il}, c_{li}\} \wedge l \in I_u^+ \wedge l \neq i, \\ -1 & \text{if } \theta \in \{c_{jl}, c_{lj}\} \wedge l \in I_u^+ \wedge l \neq j, \\ 0 & \text{else} \end{cases}$$