Recomendación Musical

IIC3633

PROFESOR: DENIS PARRA

ALUMNO: MIGUEL FADIĆ

Papers

- McFee, B., Barrington, L., & Lanckriet, G. (2012). Learning content similarity for music recommendation. *Audio, Speech, and Language Processing, IEEE Transactions on*, *20*(8), 2207-2218.

- Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., & Jambor, T. (2012, February). Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining* (pp. 13-22). ACM.

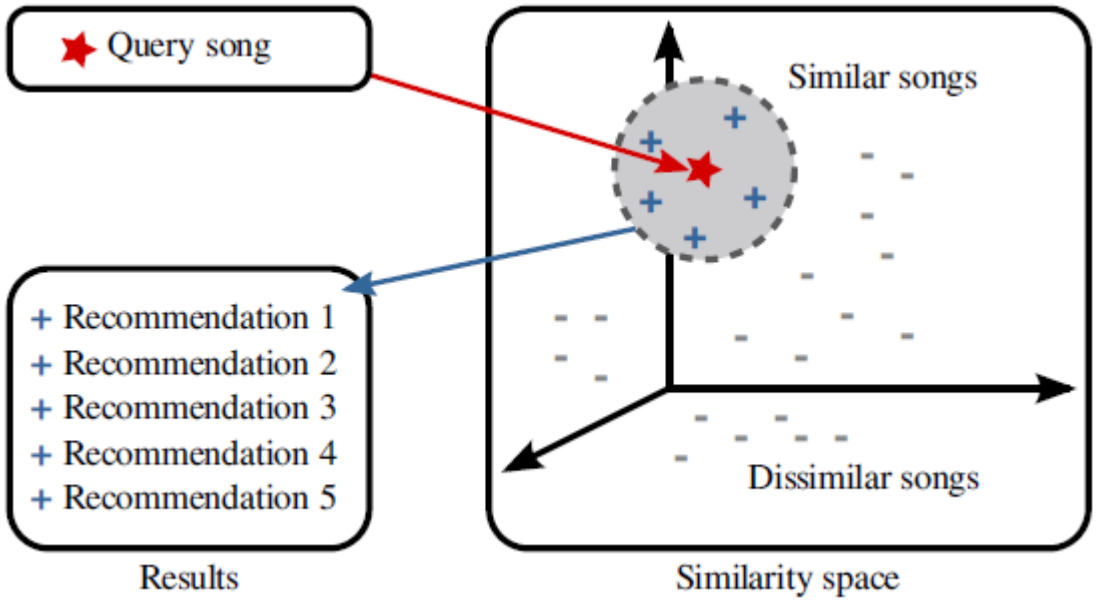
- Aizenberg, N., Koren, Y., & Somekh, O. (2012, April). Build your own music recommender by modeling internet radio streams. In *Proceedings of the 21st international conference on World Wide Web* (pp. 1-10). ACM.

Learning content similarity for music recommendation

- Contexto
- Problema
- Solución propuesta
- Experimentos
- Resultados
- Conclusiones

Contexto

- Recomendación y generación de lista cae naturalmente dentro del concepto de *query-by-example*
- Estado del arte utiliza *Collaborative Filtering* (**CF**) para representar los items musicales



Problema

- **CF** no funciona bien cuando hay poca información sobre un item

Solución propuesta

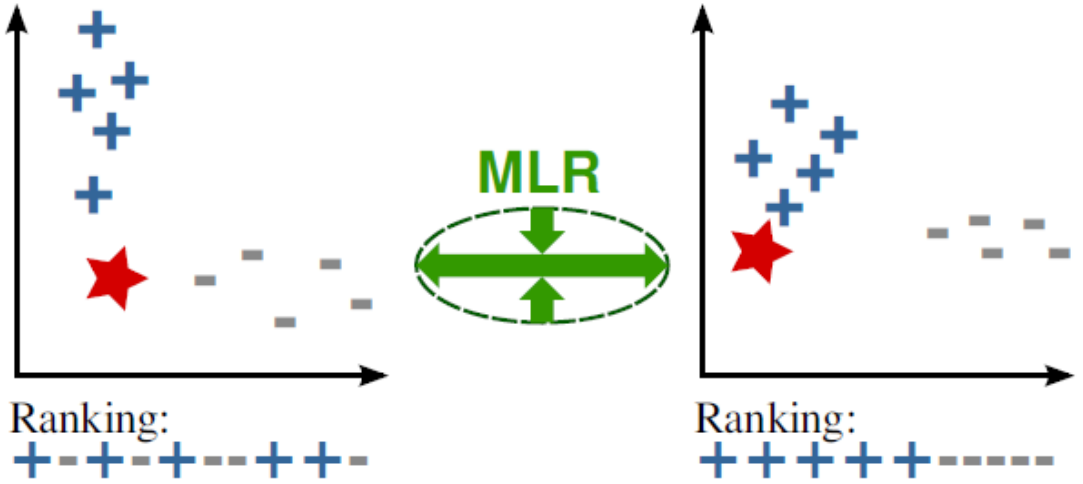
- Desarrollar una métrica de similaridad basada en contenido utilizando los datos de CF

Aprendiendo similaridad

- Dada una canción de *query* q, retornar una lista ordenada desde una base de datos *X* de n canciones según similaridad con q
- Asumimos que cada canción está representada por un vector en $\mathbb{R}^d$
- Se puede usar distancia euclideana $\|q - x\|$.

- Se puede aprender una matriz de distancia $W \in \mathbb{R}^{d \times d}$
- Distancia queda definida como

$$\|q - x\|_W = \sqrt{(q - x)^\intercal W (q - x)}. \qquad (1)$$



Similaridad basada en CF binario

- Dada *F,* una matriz de **CF** binaria (consume ítem o no lo consume)
- Se puede definir la similaridad entre los ítems *i* y *j* como

$$S(i,j) = \frac{|F[\cdot i] \cap F[\cdot j]|}{|F[\cdot i] \cup F[\cdot j]|} = \frac{F[\cdot i]^{\mathsf{T}} F[\cdot j]}{|F[\cdot i]| + |F[\cdot j]| - F[\cdot i]^{\mathsf{T}} F[\cdot j]}, \quad (2)$$

- Permite, para cada q, separar los ítems en el conjunto de los *k* más relevantes $\mathcal{X}_q^+$ y el resto $\mathcal{X}_q^-$

Se busca resolver el siguiente problema

$$\forall q: \ \langle W, \psi(q, y_q) \rangle_{\mathrm{F}} \geq \langle W, \psi(q, y) \rangle_{\mathrm{F}} + \Delta(y_q, y) - \xi_q. \quad (3)$$

- $\Delta(y_q, y)$ pérdida de predecir y en vez de y_q
- $$\psi(q, y) = \sum_{i \in \mathcal{X}_q^+} \sum_{j \in \mathcal{X}_q^-} y_{ij} \frac{(\phi(q, i) - \phi(q, j))}{|\mathcal{X}_q^+| \cdot |\mathcal{X}_q^-|}, \quad (4)$$
- $$y_{ij} = \begin{cases} +1 & i \text{ before } j \text{ in } y \\ -1 & i \text{ after } j \end{cases}, \qquad \phi(q, i) = -(q - i)(q - i)^{\mathsf{T}},$$

$$\begin{aligned}\langle W, \phi(q,i)\rangle_{\mathrm{F}} &= -\operatorname{tr}\left(W(q-i)(q-i)^{\mathsf{T}}\right) \\ &= -(q-i)^{\mathsf{T}} W (q-i) \\ &= -\|q-i\|_W^2 .\end{aligned} \tag{8}$$

Finalmente

Algorithm 1 Metric learning to rank [14]

Input: data $\mathcal{X} = \{q_1, q_2, \ldots, q_n\} \subset \mathbb{R}^d$,
correct rankings $\{y_q : \ q \in \mathcal{X}\}$,
slack trade-off $C > 0$

Output: $d \times d$ matrix $W \succeq 0$

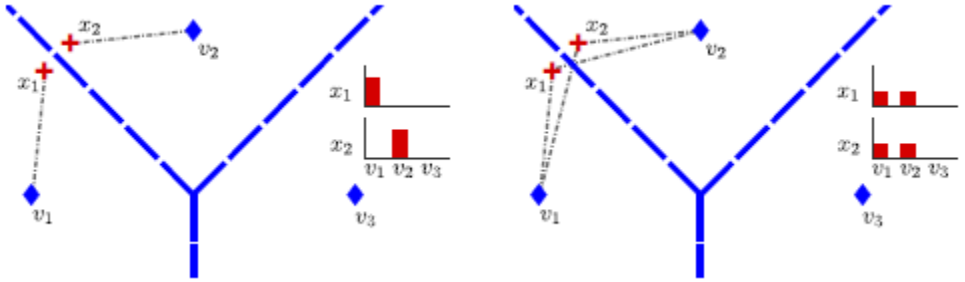
$$\min_{W \succeq 0, \xi} \quad \operatorname{tr}(W) + C \cdot \frac{1}{n} \sum_{q \in \mathcal{X}} \xi_q$$

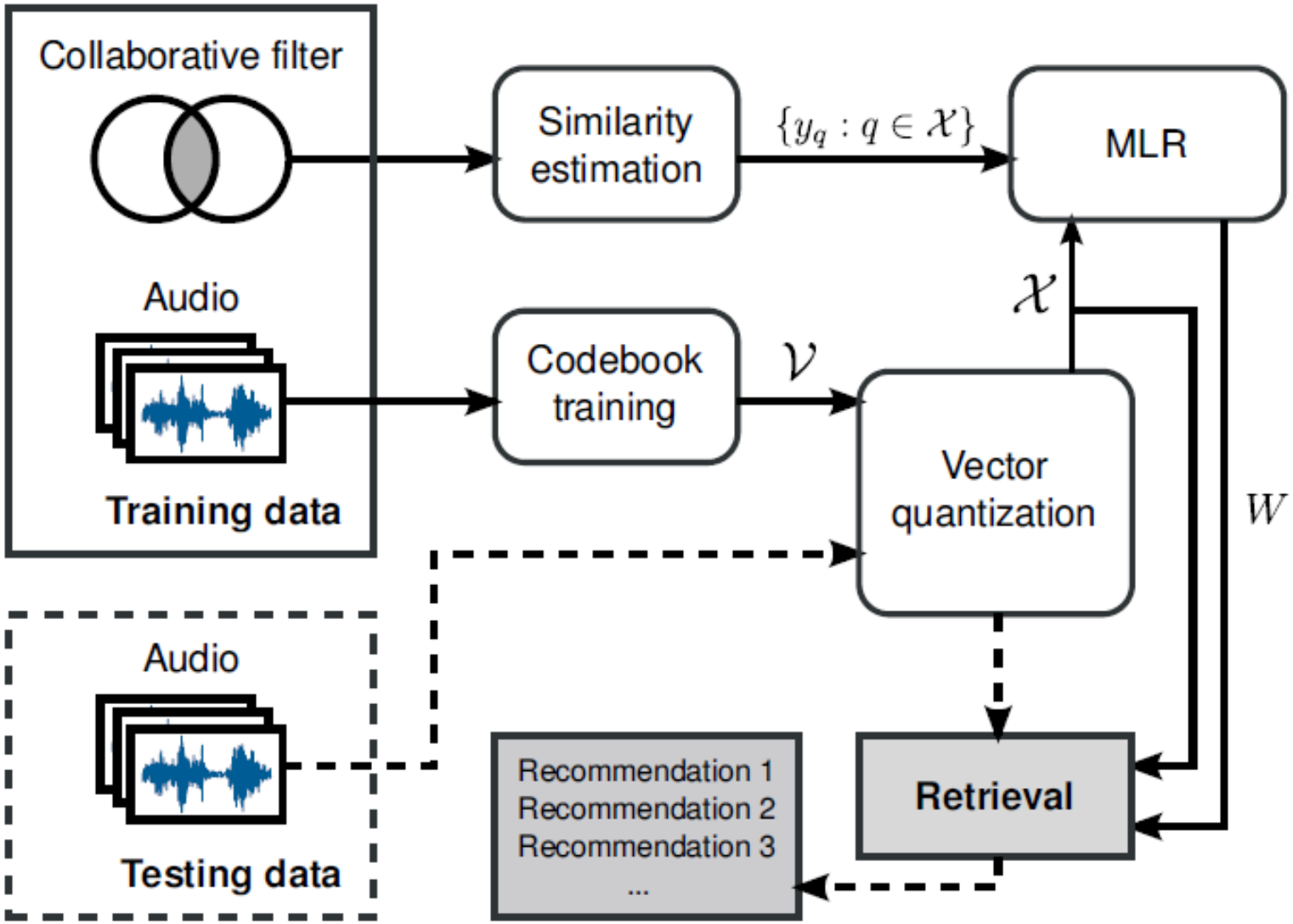
$$\text{s.t.} \quad \forall q \in \mathcal{X}, \ \forall y \in \mathcal{Y} :$$
$$\langle W, \psi(q, y_q) \rangle_{\mathrm{F}} \geq \langle W, \psi(q, y) \rangle_{\mathrm{F}} + \Delta(y_q, y) - \xi_q$$

Representando el audio

- Codebook Training
  - Mel frequency cepstral coefficients (MFCCs)
  - Codewords $\mathcal{V}$ usando k-means
- Vector quantization

$$h_x^\tau[v] = \frac{1}{|x|} \sum_{x_i \in x} \frac{1}{\tau} \mathbb{1}\left[v \in \underset{u \in \mathcal{V}}{\operatorname{argmin}}^{\tau} \|x_i - u\| \right]. \qquad (11)$$




Collaborative filter
Audio
Training data
Similarity estimation
$\{y_q : q \in \mathcal{X}\}$
MLR
$\mathcal{X}$
Codebook training
$\mathcal{V}$
Vector quantization
W
Audio
Testing data
Recommendation 1
Recommendation 2
Recommendation 3
...
Retrieval

Experimentos

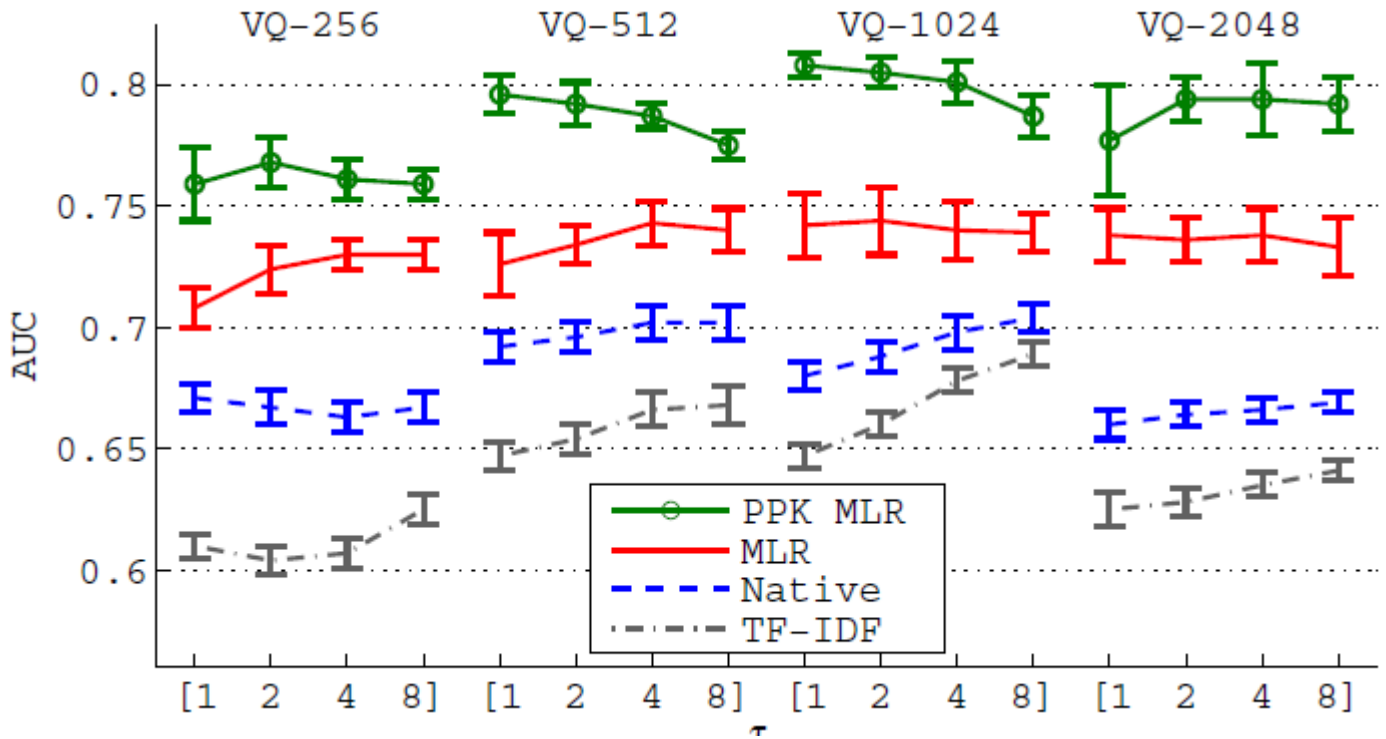
- Data
  - Collaborative filter: Last.FM
    - 359347 usuarios
    - 186642 artistas
    - Un artista es relevante para un usuario si fue escuchado al menos 10 veces
  - Audio: CAL10K
    - 4661 artistas
    - 10832 canciones
      - Experiment set
        - 2015 artistas con al menos 100 “escuchadores”
      - Codebook set
        - 2646 artistas con los artistas restantes

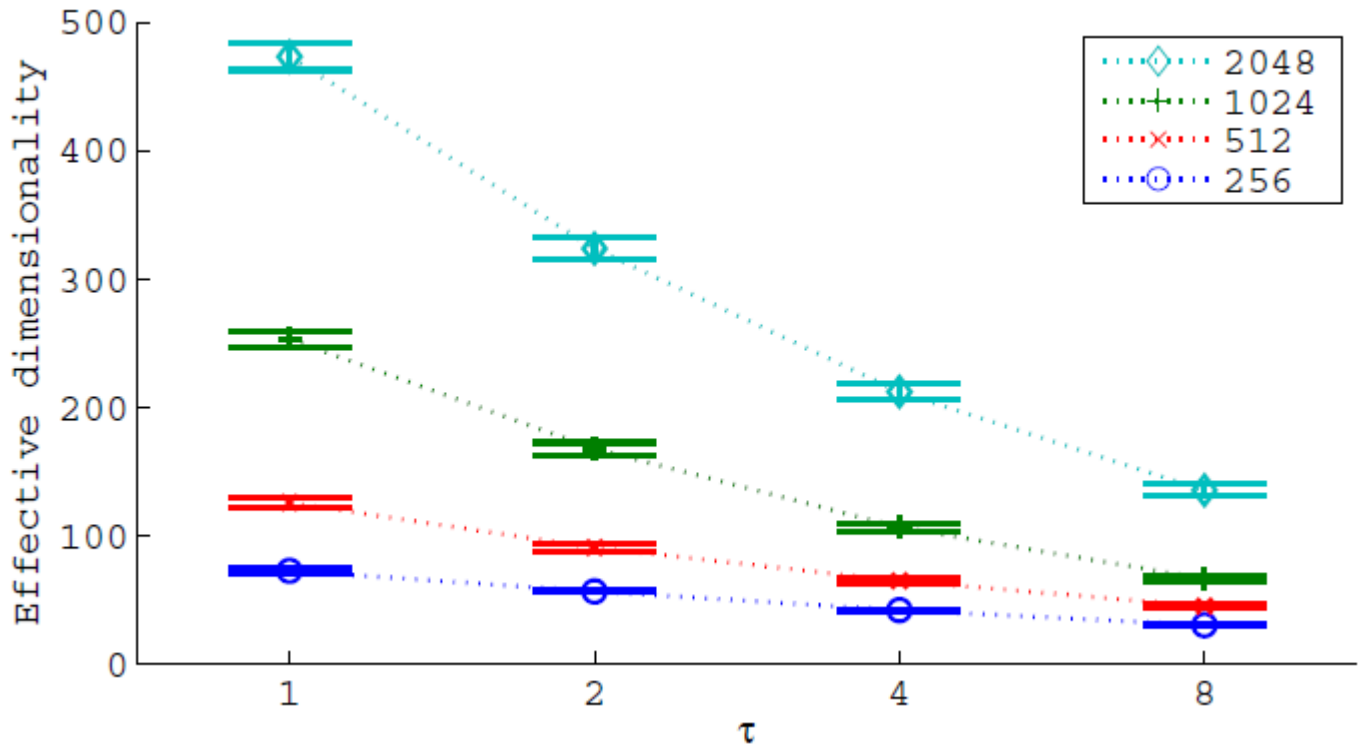
	Training	Validation	Test
# Artists	806	604	605
# Songs	2122.3 ± 36.3	1589.3 ± 38.6	1607.5 ± 64.3
# Relevant	36.9 ± 16.4	36.4 ± 15.4	37.1 ± 16.0

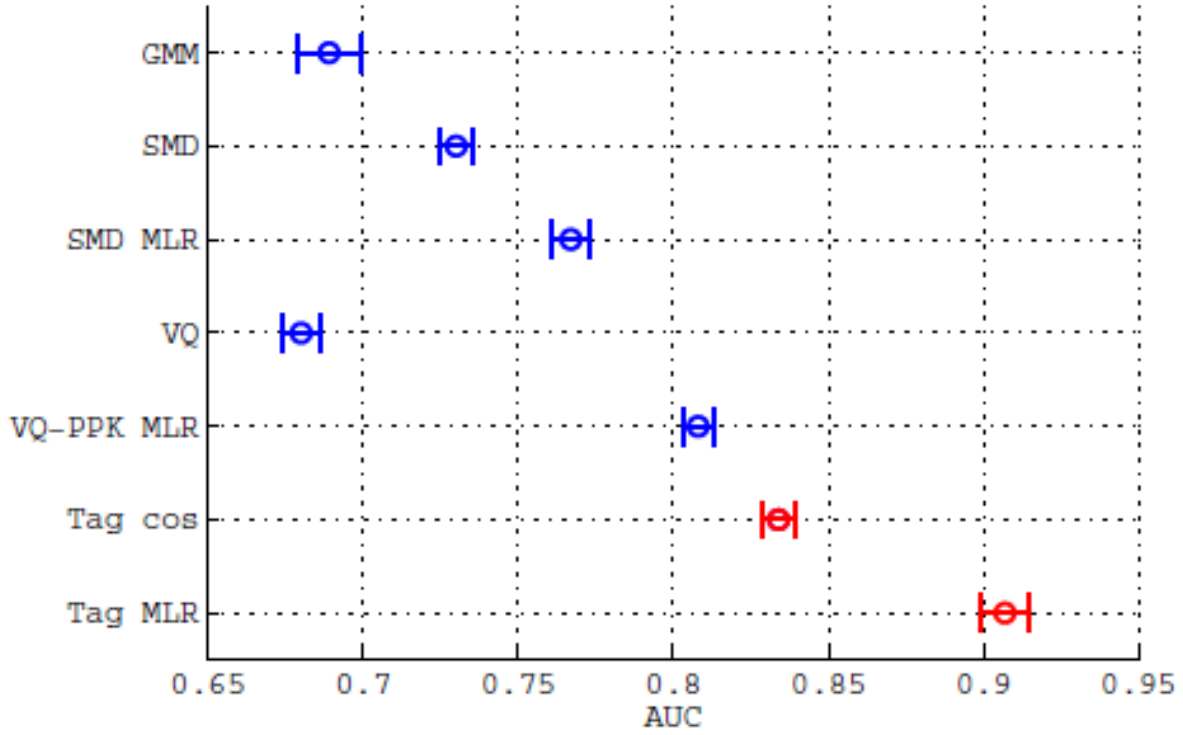
- Se probaron valores para C entre $\{10^{-2}, 10^{-1}, \cdots, 10^{9}\}$
- Para la pérdida se evaluaron
  - AUC
  - NDCG
  - MRR

- Comparación contra:
  - Gaussian mixture models
  - Unsupervised weighting of VQ codewords
  - Automatic semantic annotation
  - Manual semantic annotation

Resultados


VQ-256
VQ-512
VQ-1024
VQ-2048
AUC
0.8
0.75
0.7
0.65
0.6
[1 2 4 8]
[1 2 4 8]
[1 2 4 8]
[1 2 4 8]
PPK MLR
MLR
Native
TF-IDF


Effective dimensionality
500
400
300
200
100
0
1
2
4
8
τ
2048
1024
512
256


GMM
SMD
SMD MLR
VQ
VQ-PPK MLR
Tag cos
Tag MLR
0.65
0.7
0.75
0.8
0.85
0.9
0.95
AUC

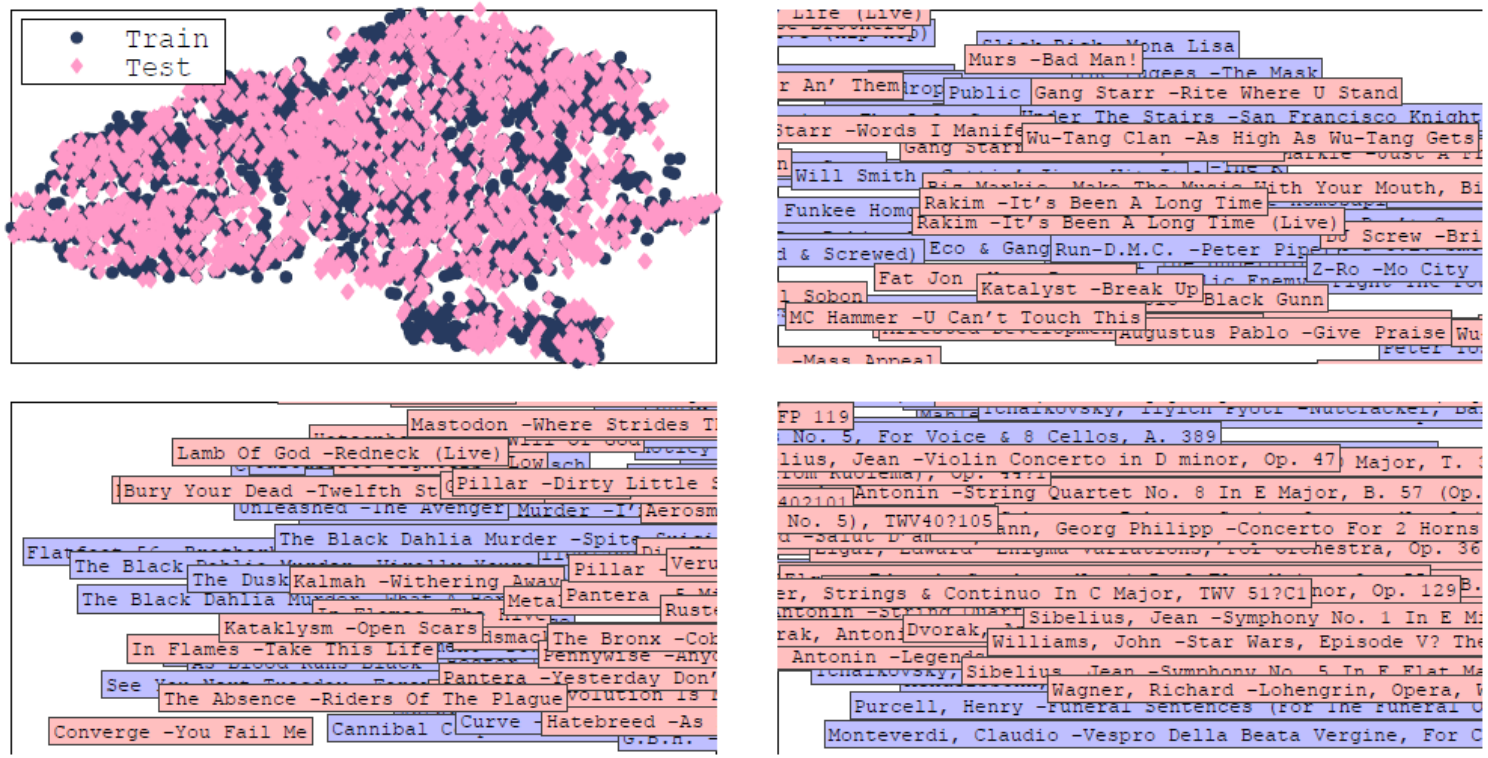


Fig. 7. A t-SNE visualization of the optimized similarity space produced by PPK+MLR on one training/test split of the data ($|\mathcal{V}| = 1024$, $\tau = 1$). Close-ups on three peripheral regions reveal *hip-hop* (upper-right), *metal* (lower-left), and *classical* (lower-right) genres.

Test query	VQ (Native)	VQ (PPK+MLR)
Ornette Coleman - Africa is the Mirror of All Colors	Judas Priest - You've Got Another Thing Comin'	Wynton Marsalis - Caravan
	Def Leppard - Rock of Ages	►Dizzy Gillespie - Dizzy's Blues
	KC & The Sunshine Band - Give it Up	►Michael Brecker - Two Blocks from the Edge
	Wynton Marsalis - Caravan	►Eric Dolphy - Miss Ann (live)
	Ringo Starr - It Don't Come Easy	Ramsey Lewis - Here Comes Santa Claus
Fats Waller - Winter Weather	►Dizzy Gillespie - She's Funny that Way	Chet Atkins - In the Mood
	Enrique Morente - Solea	►Charlie Parker - What Is This Thing Called Love?
	Chet Atkins - In the Mood	►Bud Powell - Oblivion
	Rachmaninov - Piano Concerto #4 in Gmin	►Bob Wills & His Texas Playboys - Lyla Lou
	Eluvium - Radio Ballet	►Bob Wills & His Texas Playboys - Sittin' On Top Of The World
The Ramones - Go Mental	Def Leppard - Promises	►The Buzzcocks - Harmony In My Head
	►The Buzzcocks - Harmony In My Head	Motley Crue - Same Ol' Situation
	Los Lonely Boys - Roses	►The Offspring - Gotta Get Away
	Wolfmother - Colossal	►The Misfits - Skulls
	Judas Priest - Diamonds and Rust (live)	►AC/DC - Who Made Who (live)

Conclusiones

- Es possible utilizar implicit feedback para obtener datos de entrenamiento de alta calidad sin la participación activa del usuario
- La representación propuesta permite describir de manera eficiente y compacta el contenido acústico
- Utilizando esta representación junto con una métrica de distancia optimizada se obtienen resultados sustancialmente superiores a los de otros métodos basados en contenido

Auralist: Introducing Serendipity into Music Recommendation

- Contexto
- Problema
- Solución propuesta
- Experimentos
- Resultados
- Conclusiones

Contexto

- Un sistema recomendador ideal debería ser similar a un amigo en quien confías o un experto, y entregarte una lista que balancee *accuracy*, *diversity*, *novelty* y *serendipity*

- Hasta ahora, la mayoría de los sistemas se ha preocupado de mejorar el accuracy

Problema

- Este approach no considera el deseo humano de variadad, descubrimiento y cambio

- El usuario podría quedarse atrapado en un ciclo de auto-reforzamiento

Solución propuesta

- Un framework que simultáneamente inyecta *novelty*, *diversity* y *serendipity* al proceso de recomendación

Nomenclatura

S	Set of all users
P	Preference matrix, where $P_{i,u}$ is the preference given by user u for item i (with our boolean preference system, $P_{i,u} = 1$ if the item is preferred by the user, and 0 otherwise)
R	*Top-N* function, where $R_{u,n}$ gives the top n recommended items for user u
N	Set of all items
W_u	Withheld item history of user u (set not used for training)
H_u	Non-withheld item history of user u
pop_i	Fraction of (all) preferences directed at item i
T	Set of *LDA* topics (the number of topics is 200).
L	*LDA* item-topic matrix, where $L_{i,t}$ represents the composition proportion assigned to topic t for item i
C_i	Number of artist i's unique listeners

Diversity

Representa la varidad presente en una lista de recomendaciones

$$\overline{Intra\text{-}List\ Similarity} = \frac{1}{|S|}\sum_{u \in S} \sum_{i,j \in \mathrm{R}_{u,20}, j<i} CosSim(i,j)$$

$$CosSim(i,j) = \frac{\#\text{ users who like both } i \text{ and } j}{\sqrt{\#\text{ prefs in } i} \times \sqrt{\#\text{ prefs in } j}}$$

Novelty

Puede verse como la habilidad de introducir items que el usuario no ha experimentado anteriormente en la vida real

$$\overline{Novelty} = \frac{1}{|S|} \sum_{u \in S} \sum_{i \in \mathrm{R}_{u,20}} \frac{\log_2 pop_i}{20}$$

Serendipity

- Representa lo inusual o sorpresivo de las recomendaciones
- Tiene que ver con el contenido semántico de los items

$$\overline{Unserendipity} = \sum_{u \in S} \frac{1}{|S||H_u|} \sum_{h \in H_u} \sum_{i \in R_{u,20}} \frac{CosSim(i,h)}{20}$$

Auralist Framework

- Basic Auralist
- Community-Aware
- Bubble-Aware
- Full

Basic Auralist

- LDA sobre los artistas

$$LDASim(i,j) = \frac{\sum_{t \in T} L_{i,t} \times L_{j,t}}{\sqrt{\sum_{t \in T} (L_{i,t})^2} \sqrt{\sum_{t \in T} (L_{j,t})^2}}$$

$$Basic(u,i) = \sum_{h \in H_u} LDASim(i,h)$$

Community-Aware

$$Listener\ Diversity(i) = -\sum_{t \in T} L_{i,t} \log_2(L_{i,t})$$

$$Community(u,i) = (1-\lambda) rank_{Basic,u,i} + \lambda rank_{Diversity,i}$$

$$Listener\ Diversity'(i) = Listener\ Diversity(i) - Offset_{pop}(i)$$

$$Offset_{pop}(i) = 0.462 \log(C_i) - 1.326.$$

Bubble-Aware

$$Bubble(u,i) = (1-\lambda) rank_{Basic,u,i} + \lambda rank_{Declustering,u,i}$$

```
foreach item pair j, k in H_u do
    total+= LDASim(j, k);
    count+=1;
end
avgSim = total/count;
foreach item i in candidate set for user u do
    foreach item pair j, k in neighbours(i) do
        if LDASim(j, k) > avgSim then
            edgeTotal+=1;
        end
        edgeCount+=1;
    end
    cluster(i) = edgeTotal/edgeCount;
    recommendations.add(cluster(i), i);
end
recommendations.sortBy(cluster, ascending);
```

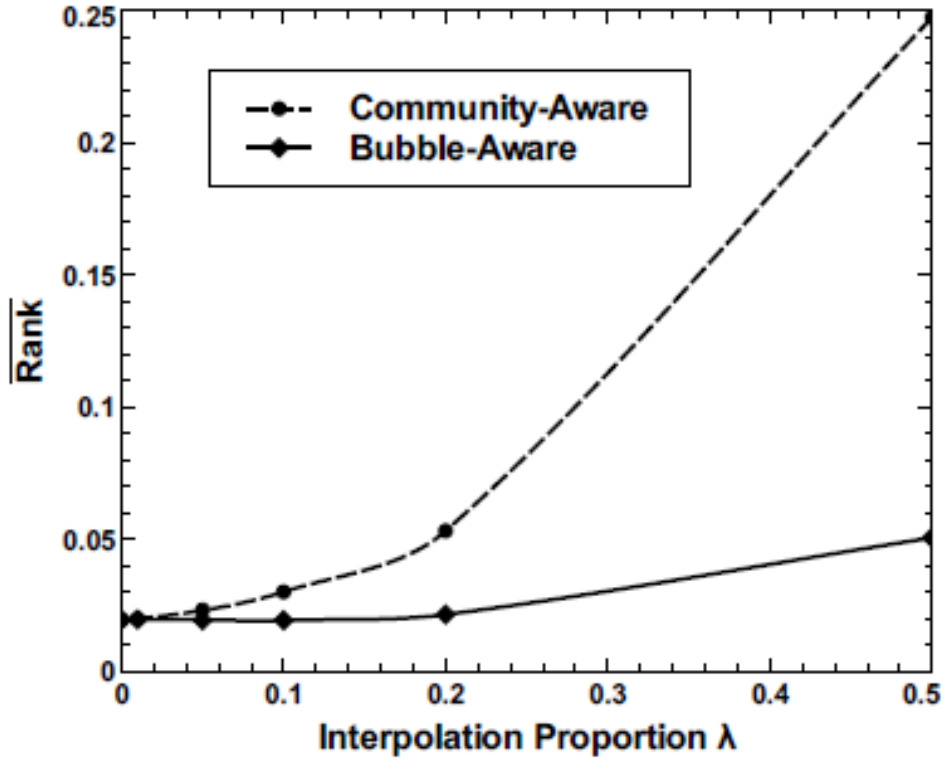
Algorithm 1: Pseudo-code for *Declustering*.

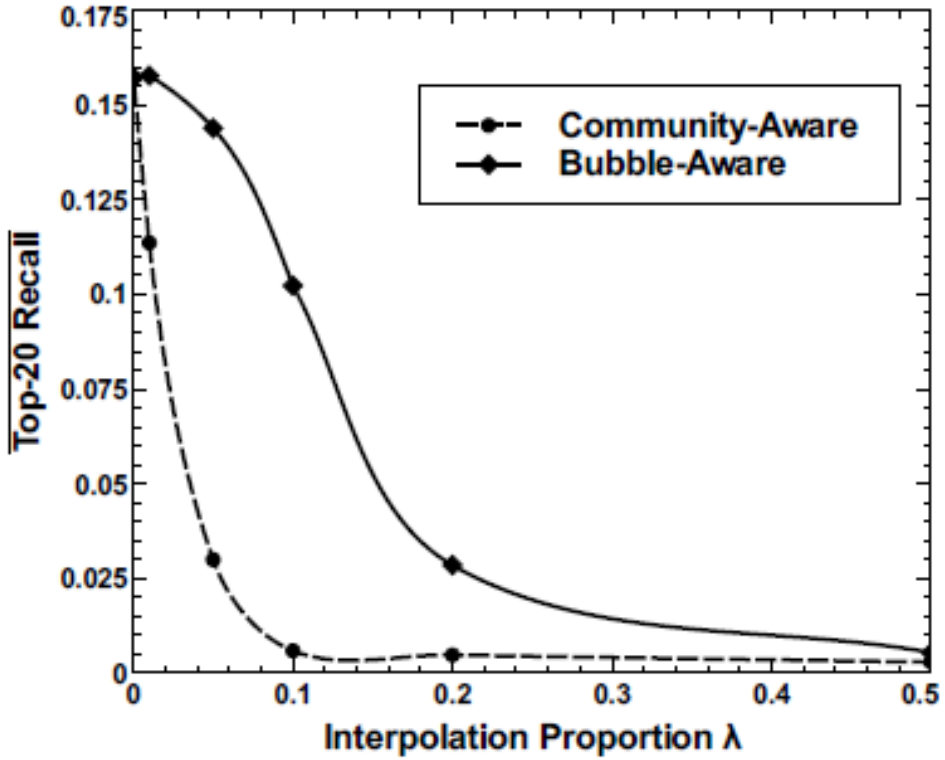
Experimentos

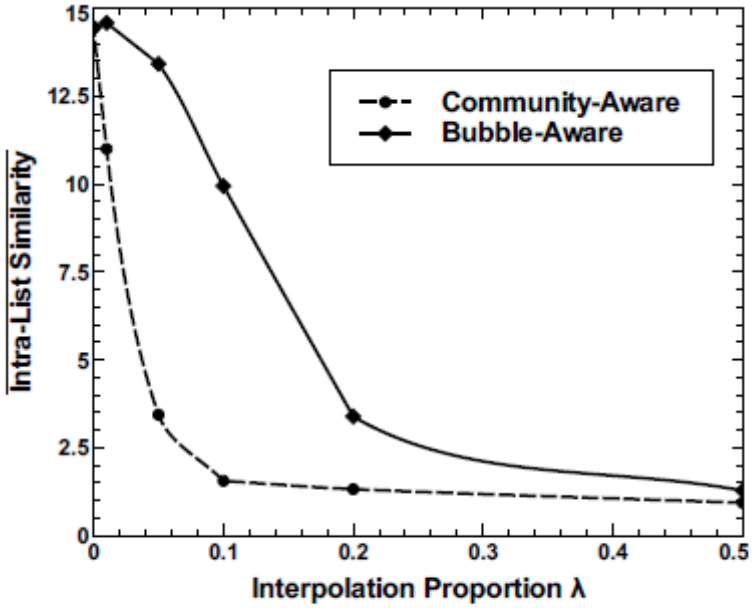
- Data
  - Last.fm
    - 360k usuarios
    - 48988 artistas para recomendar

Resultados

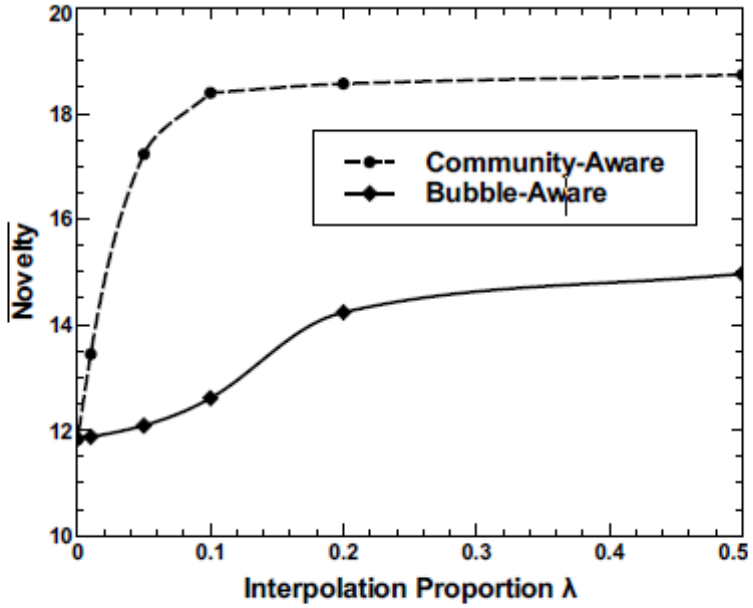
	$\overline{Rank}$	$\overline{Top\text{-}20\ Recall}$	$\overline{Intra\text{-}List\ Similarity}$	$\overline{Novelty}$	$\overline{Unserendipity}$
Basic Auralist	**0.019** ±0.0004	0.157 ±0.004	**14.4** ±0.2	**11.8** ±0.06	0.060 ±0.0004
Implicit SVD	0.039 ± 0.0008	**0.174** ±0.002	14.7 ±0.1	10.9 ±0.03	**0.046** ±0.0002
Community-aware(λ=0.05)	0.023 ±0.02	0.030 ±0.0009	3.4 ±0.06	17.2 ±0.1	0.047 ±0.0003
Bubble-aware(λ=0.2)	0.021 ±0.0002	0.029 ±0.0006	3.4 ±0.05	14.2 ±0.1	0.035 ±0.0002
Full Auralist	0.025	0.008	1.54	17.3	0.039


0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
Community-Aware
Bubble-Aware
Rank
Interpolation Proportion λ

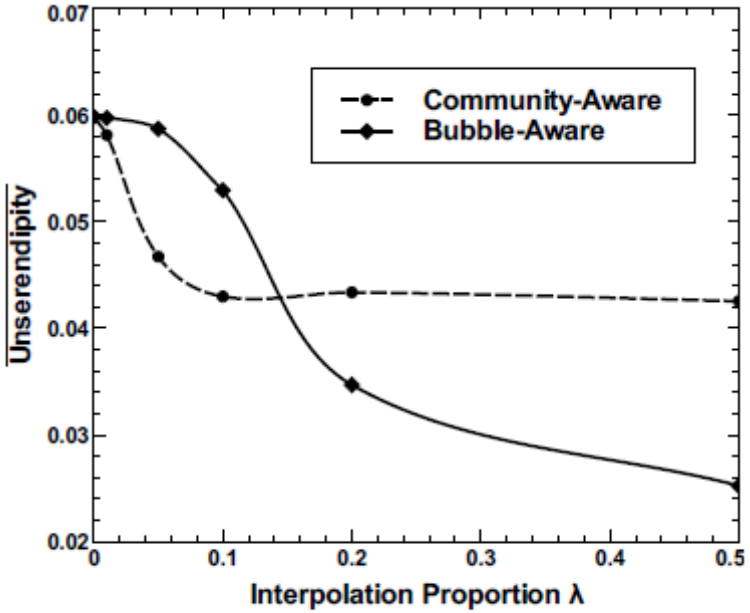

0.175
0.15
0.125
0.1
0.075
0.05
0.025
0
0
0.1
0.2
0.3
0.4
0.5
Community-Aware
Bubble-Aware
Top-20 Recall
Interpolation Proportion λ



(a) Diversity



(b) Novelty

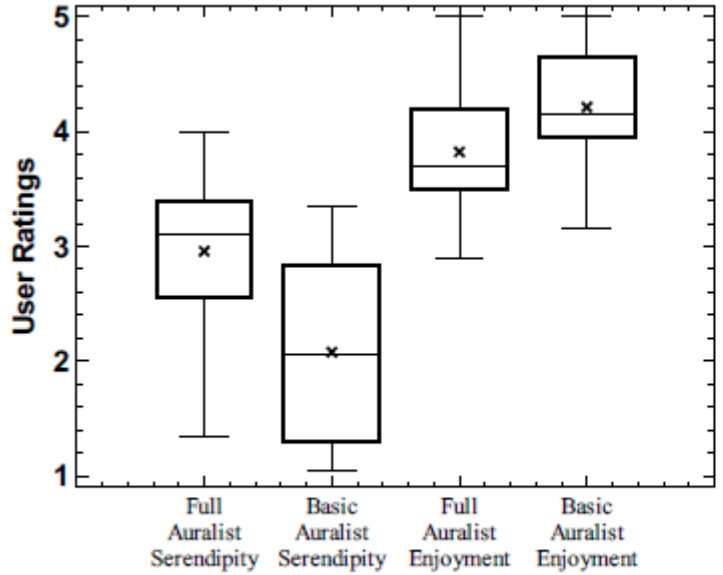


(c) Serendipity

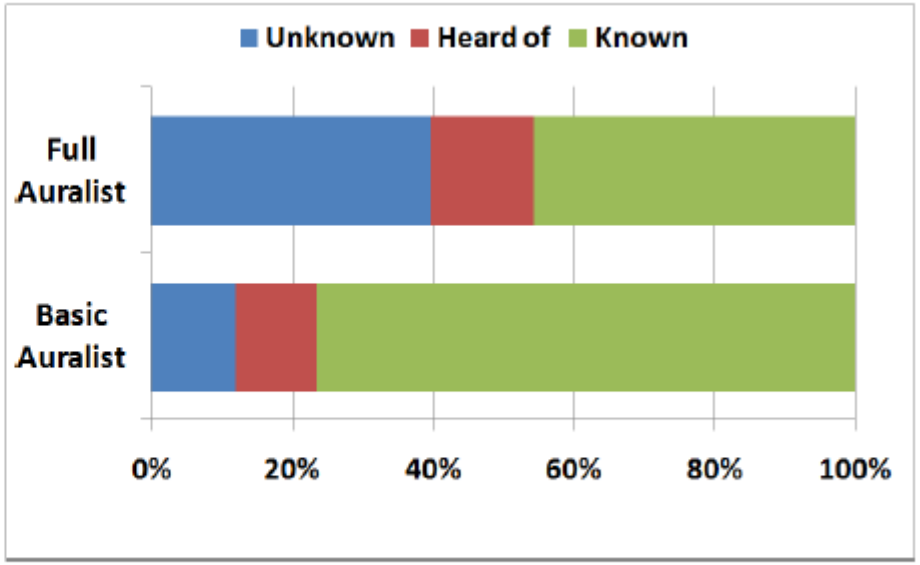
User study

- 21 participantes
- 6 artistas pre 2008 como seed por usuario

	Basic Auralist	Full Auralist
Serendipity Rating	2.08 (±0.77)	2.96 (±0.69)
Enjoyment Rating	4.21 (±0.54)	3.82 (±0.53)
# Useful Recommendations	2.90 (±2.61)	5.86 (±3.05)
# Serendipitous Recomm.s	1.81 (±1.86)	4.14 (±2.90)
# Familiar Recomm.s	12.62 (±4.14)	7.43 (±5.07)



(a) *Serendipity* and *Enjoyment* ratings



(b) Fraction of novel recommendations

Conclusiones

- Es posible aumentar *novelty*, *diversity* y *serendipity* sin comprometer demasiado el *accuracy*

Build Your Own Music Recommender by Modeling Internet Radio Streams

- Contexto
- Problema
- Solución propuesta
- Experimentos
- Resultados
- Conclusiones

Contexto

- Collaborative Filtering es una técnica ampliamente utilizada por los sistemas recomendadores

- Es capaz de explotar *Popularity Trends* que pasarían totalmente desapercibidos por técnicas *Content Based*

Problema

- Se requiere gran cantidad de datos históricos

- “Rich get richer”
  - Sistemas ya establecidos aumentan sus datos
  - Barrera de entrada para sistemas nuevos

Solución

- Utilizar los *Internet Radio Streams* como fuente de datos

- Ventajas
  - Freshness
  - Completeness
  - Robustness
  - Scale
  - Diversity

Modelando las playlist

- Los *items* son canciones y artistas

  Each item i is mapped into a vector $p_i \in \mathbb{R}^\ell$.

- Las *stations* son usuarios y estaciones

  Each station s is mapped into a vector $v_s \in \mathbb{R}^\ell$.

- El arista que toca la canción i se denota como $a(i)$
- El bias por la popularidad se denota $c_i \in \mathbb{R}$.
- Se utiliza una representación de factores latentes
  - item $q_i \stackrel{\text{def}}{=} p_i + p_{a(i)}$
  - bias $b_i \stackrel{\text{def}}{=} c_i + c_{a(i)}$

- Se define la afinidad del item *i* con la estación *s* en el instante *t* como $r_{si;t}$

- Se modela el likelihood de observar a *i* como un item tocado en la estación *s* en el momento *t* como la distribución multinomial

$$P(i|s;t,\Theta) = \frac{\exp(r_{si;t})}{\sum_j \exp(r_{sj;t})}$$

- Una primera aproximación de la afinidad podría ser

$$r_{si;t} \stackrel{\text{def}}{=} b_i + q_i^T v_s$$

- Una version que considera los distintos momentos del día, además de la “vecindad” de las canciones tocadas resulta en

$$r_{si;t} \stackrel{\text{def}}{=} b_i + q_i^T \left(v_s + v_s^{(\text{slot}(t))} + \frac{1}{\sqrt{|P_s^{(t,w)}|}} \sum_{j \in P_s^{(t,w)}} q_j \right)$$

Donde $P_s^{(t,w)}$ es el conjunto de todas las canciones tocadas por la estación s entre [*t*-*w*,*t*)

- Finalmente se busca maximizar el log-likehood

$$L(\mathcal{S};\Theta) \stackrel{\text{def}}{=} \sum_{P_s \in \mathcal{S}} \sum_{(i,t) \in P_s} \log P(i|s;t,\Theta)$$

Agregando nuevos usuarios

- Se puede reemplazar a v_s por $|P_s|^{-0.5} \sum_{j \in P_s} q_j$
  - Obteniendo un obteniendo un modelo "*station-less*"

$$r_{si;t} \overset{\text{def}}{=} b_i + \left(q_i^{(1)}\right)^T \left(\frac{1}{\sqrt{|P_s|}} \sum_{j \in P_s} q_j^{(2)} + \frac{1}{\sqrt{|P_s^{(t,w)}|}} \sum_{j \in P_s^{(t,w)}} q_j^{(3)} \right)$$

$$r_{si;t} = b_i + q_i^T \left(\frac{1}{\sqrt{|P_s|}} \sum_{j \in P_s} q_j + \frac{1}{\sqrt{|P_s^{(t,w)}|}} \sum_{j \in P_s^{(t,w)}} q_j \right)$$

Experimentos

Dataset	#playlists	#items	#plays(train)	#plays(test)	#non-repeat-plays(test)
IRDB	3,535	167,695	5,955,208	1,983,049	247,109
StrongGen1	2,649	167,185	5,967,980	N/A	N/A
StrongGen2	886	140,971	1,478,073	492,204	59,069
Yahoo! Music	203,956	41,727	18,783,943	4,596,710	4,596,710

- Métrica:
  - Misma de KDD-Cup’11 challenge
    - Para cada item en test set, se escoge un item al azar que no haya sido reproducido por el usuario
    - Se usa el modelo para rankear ambos items
    - La fracción de pares rankeados correctamente representa el valor de la métrica

- Baseline:
  - Popularidad
  - Genero
  - k-Means

Weak

Method	NonRepeat-vs-Uni	NonRepeat-vs-Pop	Played-vs-Pop
Popularity	65.12%	39.91%	50.26%
Genre	78.41%	73.89%	79.33%
k-Means	74.61%	72.83%	83.80%
station-based model	90.91%	87.38%	95.72%
station-less model	**91.41%**	**88.66%**	**95.95%**

Strong

Method	NonRepeat-vs-Uni	NonRepeat-vs-Pop	Played-vs-Pop
Popularity	65.24%	34.59%	49.94%
Genre	78.70%	74.86%	80.21%
k-Means	74.37%	72.83%	83.99%
station-based model	**90.38%**	86.96%	91.90%
station-less model	90.23%	**87.98%**	**92.56%**

Yahoo

Method	NonRepeat-vs-Uni	NonRepeat-vs-Pop
Trained-on-YMDB	*96.33%*	*82.85%*
Popularity-on-YMDB	*93.18%*	*46.95%*
Genre	78.22%	75.19%
k-Means	63.94%	59.99%
station-based model	91.25%	77.14%
station-less model	90.79%	75.13%

Conclusión

- Es posible obtener grandes resultados utilizando datos disponible públicamente