RECOMMENDER SYSTEMS :

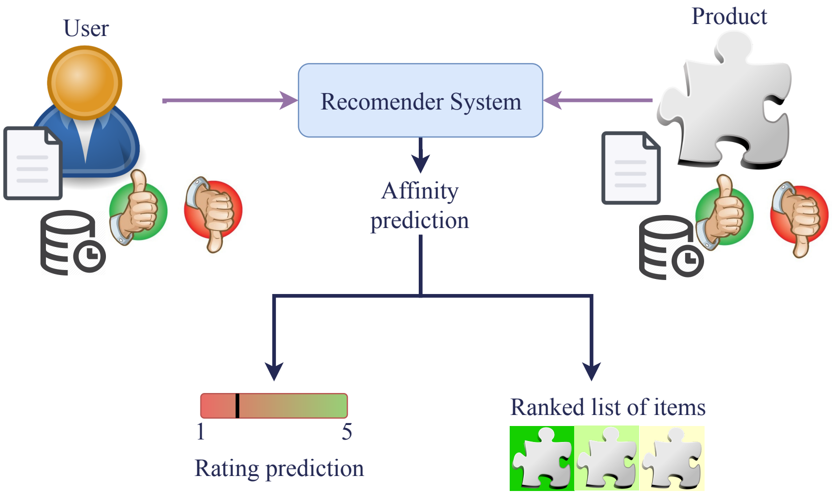
ADVANCED ARCHITECTURES
NEW METRICS

Vincent Guigue

SCIENCES SORBONNE UNIVERSITÉ
LIP6
MLIA Machine Learning & Deep Learning for Information Access

Introduction

New issues

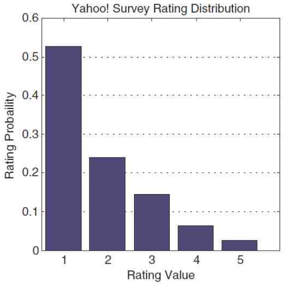

User
Product
Recomender System
Affinity
prediction
1
5
Rating prediction
Ranked list of items

Main issue : the weakness of MCAR hypothesis

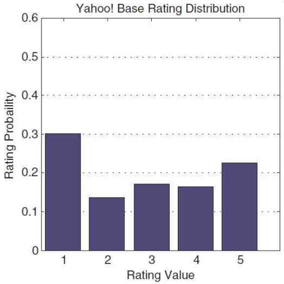
SCIENCES SORBONNE UNIVERSITÉ

Data are not Missing Completely At Random...

Graphs from [Marlin & Zemel '09]:



Survey: ask users to rate a <u>random</u> list of items: approximates **complete** data



Typical Data: users are <u>free to choose</u> which items to rate -> available data are **MNAR** :

instead of giving low ratings, users tend to not give a rating at all.

Even different in product/movie domains:

- 60-80% of 4/5 ratings

Main issue : the weakness of MCAR hypothesis

SCIENCES SORBONNE UNIVERSITÉ

Data are not Missing Completely At Random...

Predicting profile behavior on this kind of data:

H. Steck, KDD, 2010
Training and Testing of Recommender Systems on Data Missing Not at Random

Table 1: Simplistic Example for ratings missing not at random (MNAR): test data where users rated only what they liked or knew.

		users							
		horror fans				romance lovers			
	h	5		5	5				
m	o	5	5						
o	r		5		5				
v	.	5		5	5				
i	r					5	5		5
e	o						5	5	5
s	m					5		5	
	.						5	5	5

Credit: H. Steck

Main issue : the weakness of MCAR hypothesis

Data are not Missing Completely At Random...

Several outcomes:

- Changing the **error function**
  - ranking criteria
- Changing **the task**
  - predicting rated item (not the rate)

New approaches

Implicit Feedback (SVD++)

For example, a dataset shows that users that rate “Lord of the Rings 3” high also gave high ratings to “Lord of the Rings 1–2”.
$\Rightarrow$ establish high weights from “Lord of the Rings 1–2” to “Lord of the Rings 3”.
Now, if a user did not rate “Lord of the Rings 1–2” at all, his predicted rating for “Lord of the Rings 3” will be penalized.

- Binary coding : interesting / not interesting
  - positive rating / simple visit = 1
  - negative rating / missing values = 0
- Initial predictor f
  - Simple heuristic / expert knowledge...
- $R(u)$: set of items rated by $u + f$

$\Rightarrow$ $N(u)$: set of items implicitly rated by u

Yehuda Koren, KDD 2008
Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model

SVD++

General idea:

$$\hat{r}_{ui} = \underbrace{b + b_u + b_i}_{b_{ui}} + \frac{1}{\sqrt{|R(u)|}} \sum_{j \in R(u)} (r_{uj} - b_{uj}) w_{ij} + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} c_{ij}$$

Overweighting deviation for prolific users ($\frac{1}{\sqrt{|R(u)|}}$ instead of $\frac{1}{|R(u)|}$)

w_{ij} Learning deviation meaning wrt b_{uj}

c_{ij} Learning the meaning of j absence wrt i

- ... Too expensive (too many coefficients to learn)

Factorized formulation = SVD++:

$$\hat{r}_{ui} = b_{ui} + \mathbf{i} \cdot \left(\mathbf{u} + \frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} \mathbf{y}_j \right)$$

$\Rightarrow$ Y. Koren winning proposal to the Netflix challenge (2009)

Factorization machine

Back to (huge) linear model !

- Factor = interactions between items
- Easy to encode unseen effect

Feature vector **x** / Target y

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$U = \{\text{Alice (A)}, \text{Bob (B)}, \text{Charlie (C)}, \ldots\}$$

$$I = \{\text{Titanic (TI)}, \text{Notting Hill (NH)}, \text{Star Wars (SW)}, \text{Star Trek (ST)}, \ldots\}$$

$$S = \{(\text{A}, \text{TI}, \text{2010-1}, 5), (\text{A}, \text{NH}, \text{2010-2}, 3), (\text{A}, \text{SW}, \text{2010-4}, 1), (\text{B}, \text{SW}, \text{2009-5}, 4), (\text{B}, \text{ST}, \text{2009-8}, 5), (\text{C}, \text{TI}, \text{2009-9}, 1), (\text{C}, \text{SW}, \text{2009-12}, 5)\}$$

S. Rendle, ICDM 2010
Factorization machines

Bayesian Personalized Ranking

SCIENCES SORBONNE UNIVERSITÉ

u_1 prefers (i_2 to i_1) & (i_3 to i_1) $\Rightarrow +$

For each user, what are the prefered items?

$$p(i >_u j|\theta) = \frac{1}{1+\exp(-f_\theta(u,i,j))}$$

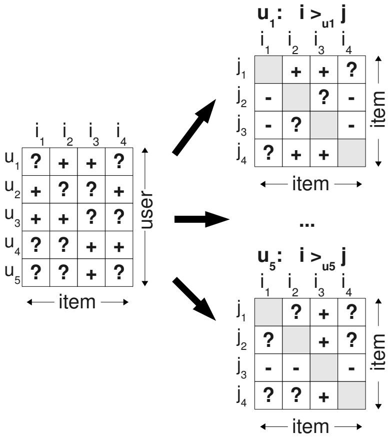
For instance (inspired from MF):

$$f_\theta(u,i,j) = \mathbf{u}\cdot\mathbf{i} - \mathbf{u}\cdot\mathbf{j}$$

NB: same nb of parameters than MF

Evaluation =

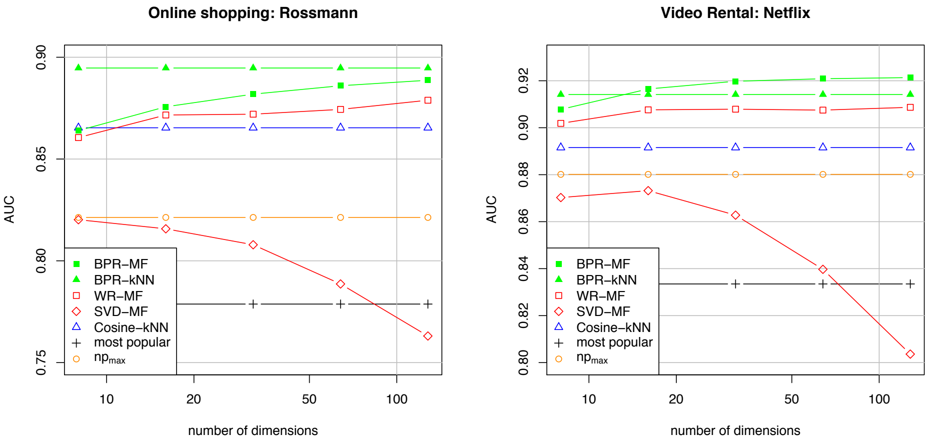
$$AUC = \frac{1}{n_u}\sum_u \frac{1}{|E(u)|}\sum_{(i,j)\in E(u)} \delta(\mathbf{u}\cdot\mathbf{i} > \mathbf{u}\cdot\mathbf{j})$$



S. Rendle et al., UAI 2009
BPR: Bayesian Personalized Ranking from Implicit Feedback

Bayesian Personalized Ranking

SCIENCES SORBONNE UNIVERSITÉ



The ranking criterion enables us to exploit high dimensional user/item representation

S. Rendle et al., UAI 2009
BPR: Bayesian Personalized Ranking from Implicit Feedback

AllRank

SCIENCES SORBONNE UNIVERSITÉ

- $r_{ui}^{(o\&i)} \Rightarrow$ All binary rating
  - 0 for missing
  - 1 relevant
- $W_{ui} = \begin{cases} 1 & \text{if observed} \\ w_m & \text{if missing} \end{cases}$
- $b_{ui}^{(o\&i)} = \begin{cases} b_{ui} & \text{if observed} \\ r_m & \text{if missing} \end{cases}$
- w_m, r_m (& λ) have to be tuned

Cost function:

$$\mathcal{L} = \sum_u \sum_i W_{ui} \left[\left(r_{ui}^{(o\&i)} - (b_{ui}^{(o\&i)} + \mathbf{u} \cdot \mathbf{i}) \right)^2 + \lambda(\|\mathbf{u}\|^2 + \|\mathbf{i}\|^2) \right]$$

H. Steck, RecSys 2010
Training and Testing of Recommender Systems on Data Missing Not at Random

Learning to rank

SCIENCES SORBONNE UNIVERSITÉ

- Pointwise :
  - Ranking score based on regression (or classification)
  - RS seen example: SVD approaches
- Pairwise :
  - Loss function is defined on pair-wise preferences
  - RankSVM, RankBoost, RankNet, FRank...
  - RS seen example: BPR
- Listwise :
  - Gradient descent on smoothed version of objective function (e.g. CLiMF presented at Recsys 2012 or TFMAP at SIGIR 2012)
  - SVM-MAP relaxes the MAP metric by adding it to the SVM constraints
  - AdaRank (modified version of Adaboost)

Evaluation:
evaluation metrics
vs
learning metrics

How to evaluate RS performance?

SCIENCES SORBONNE UNIVERSITÉ

Warning

We should not confuse **evaluation metrics** & **learning metrics**

$\Rightarrow$ MSE is a convenient learning metrics

(easily differentiable + convex ...)

... but it is a poor evaluation metrics

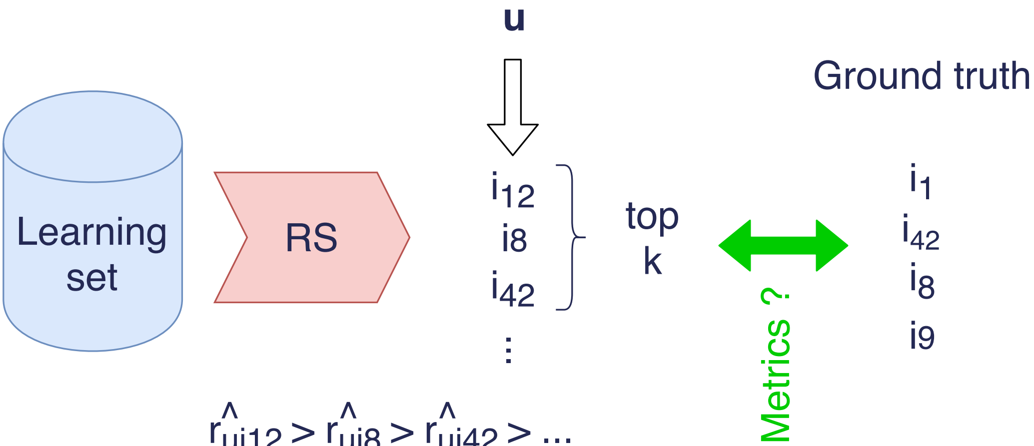
... cf Netflix Challenge feedbacks

It do not tell us if we provide relevant suggestions

- What are the other available metrics?
  - Have a look towards the IR community
- Can we use those metrics during the learning step?

Precision / Recall

SCIENCES SORBONNE UNIVERSITÉ



$$\hat{r}_{ui12} > \hat{r}_{ui8} > \hat{r}_{ui42} > \ldots$$

1/0 labeling, AUC metrics

- Rendle popularize both 1/0 prediction & AUC metrics
- AUC = tradeoff between precision & recall
  - Percentage of correct binary ranking for ONE user
  - Aggregation over n_u users

$$AUC = \frac{1}{n_u} \sum_u \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(\mathbf{u} \cdot \mathbf{i} > \mathbf{u} \cdot \mathbf{j})$$

\+ k not required

− **top** of the list = same impact as **bottom** of the list

Mean Average Precision (from the IR domain)

SCIENCES SORBONNE UNIVERSITÉ

RS aim at proposing an ordered list of suggestion...

Which **head** is far more important than the rest.

For a user u with 4 liked items to discover:

$$query = \mathbf{u} \Rightarrow RS_1 \Rightarrow \begin{bmatrix} i_{12} \\ i_8 \\ i_{42} \\ i_1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} i_1 \\ i_{42} \\ i_8 \\ i_9 \end{bmatrix} = GT$$

- Average precision (one query/user) :

$$\frac{1}{K}\sum_{k=1}^{K} precision@K = \frac{1}{4}(0 + \frac{1}{2} + \frac{2}{3} + \frac{3}{4}) = 0.478$$

- Mean Average Precision =

Averaging over the whole population

Mean Average Precision (from the IR domain)

SCIENCES SORBONNE UNIVERSITÉ

RS aim at proposing an ordered list of suggestion...

Which **head is far more important** than the rest.

For a user u with 4 liked items to discover:

$$query = \mathbf{u} \Rightarrow RS_2 \Rightarrow \begin{bmatrix} i_1 \\ i_8 \\ i_{42} \\ i_{12} \end{bmatrix} \Leftrightarrow \begin{bmatrix} i_1 \\ i_{42} \\ i_8 \\ i_9 \end{bmatrix} = GT$$

- Average precision :

$$\frac{1}{4}\sum_{k=1}^{4} precision@K = \frac{1}{4}(1 + 1 + 1 + \frac{3}{4}) = 0.9375$$

- Mean Average Precision =

Averaging over the whole population

Mean Reciprocal Rank

At which rank is the first relevant item?

$$query = \mathbf{u} \Rightarrow RS \Rightarrow \begin{bmatrix} i_{12} \\ i_8 \\ i_{42} \\ i_1 \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} i_1 \\ i_{42} \\ i_8 \\ i_9 \end{bmatrix} = GT$$

$$RR = \frac{1}{rank_i} = \frac{1}{2} \text{ on previous example}$$

Mean Reciprocal Rank = Averaging over the whole population

$\Rightarrow \approx$ *How many iterations to obtain a relevant item?*

nDCG : Normalized Discounted Cumulative Gain

SCIENCES SORBONNE UNIVERSITÉ

We assume that we have a relevance score for each item...

$$query = \mathbf{u} \Rightarrow RS \Rightarrow \left[\begin{array}{cc} i_{12} & ind = 1 \\ i_8 & ind = 2 \\ i_{42} & ind = 3 \\ i_1 & ind = 4 \end{array} \right] \Leftrightarrow \begin{bmatrix} 0 \\ 2 \\ 3 \\ 3 \end{bmatrix} = relevance$$

$$\mathrm{DCG_p} = \sum_{ind=1}^{p} \frac{relev_{ind}}{\log_2(ind+1)} = 0 + 1.26 + 1.5 + 1.29 = 4.05$$

$$\mathrm{nDCG} = \frac{\mathrm{DCG}}{\mathrm{IdealDCG}} = \frac{4.05}{3 + 1.89 + 1 + 0/0.86} = 0.69/0.6$$

Relative ideal (among suggestions) *vs* Absolute ideal (among all items)

ATOP

recall@k =

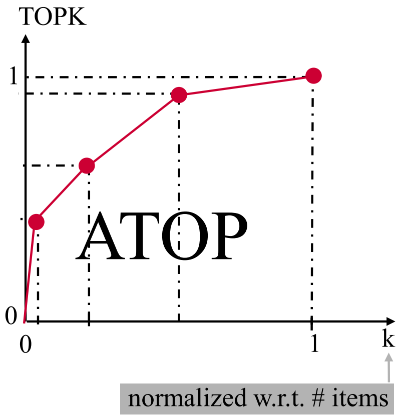
$$\frac{\#\text{relevant items in top k}}{\#\text{relevant items}}$$

Compute all recall@k...

until k match $R(u)$

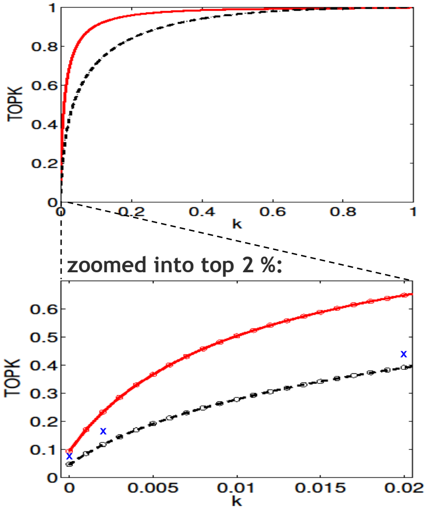
Compute the area under the curve

- focus on rated items
- numerical indicator + graphical details



H. Steck, KDD, 2010
Training and Testing of Recommender Systems on Data Missing Not at Random

ATOP



Comparison of Approaches:

— AllRank (RMSE = 1.106)

- - ignore missing ratings (RMSE = 0.921)

x integrated model [Koren '08] (RMSE = 0.887) (trained to minimize RMSE)

Large increase in Top-k Hit-Rate when accounting also for missing ratings when training on MNAR data.

39 % .... 50 % larger Top-k Hit-Rate: AllRank vs. integrated model

A/B testing & production launch

In a real situation

Designing an online Recommender System offers new performance indicators

- Online click, purchase, etc

A/B **testing:**

1. Defining some performance indicator with expert
2. Re-direct a small part of the customers to the new system B
   - make sure that the redirection is random (not biased)
3. Compare indicators from A and B

$\Rightarrow$ **Best evaluation**...

But only available **online** & with **access to the backoffice**

Serendipity : another important factor to evaluate...

SCIENCES SORBONNE UNIVERSITÉ

... But very difficult to quantify

- Exploration / exploitation dilemma
- Clustering / categorization exploitation
    - propose items from different region
- Post processing / HMI issue

CF can offer serendipity

- increase neighborhood,
- increase implicit feedback weight
- ...

Idea to design a metric

1. Learn a strong baseline (SVD)
2. New system RS
3. Unexpectedness = RS\SVD
4. Serendipity = usefulness(Unexpectedness)

CB is not well adapted

- Clustering heuristics
- bad performance

M. Ge et al., RecSys, 2010
Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity

Conclusion

Conclusions

- For many applications such as Recommender Systems (but also Search, Advertising, and even Networks) understanding data and users is vital
- Algorithms can only be as good as the data they use as input
- But the inverse is also true: you need a good algorithm to leverage your data
- Importance of User/Data Mining is going to be a growing trend in many areas in the coming years
- RS have the potential to become as important as Search is now

$\Rightarrow$ there are still many open questions and a lot of interesting research to do!

Performance = industrial global deployment

Navigation traces & user generated contents = behavior sensors

- Collecting the data
  - Storing many trace for further exploitation
    - Large amount of data = cost
  - ... But trying to reduce the **noise**
    - missing/corrupted data, inadvertently operations...
  - ... and extract implicit **feedback** (and/or specific features)
    - e.g. video watching statistics
- UI integration
  - 50% $\Rightarrow$ 90% of the job (Netflix !)
- ROI: return on invest
- Other expected benefits:
  - Fighting against **adversarial noise**
    - spam, web spam, review spam