Graphs in ML Project Defence
Recommender system with serendipity

Clémence Réda

École Normale Supérieure Paris-Saclay

January 17^{th}, 2019

Introduction Field of research

Comic from artist Piccolo

This might be a useful recommender system!

Introduction Field of research

(Adapted) comic from artist Piccolo

This might be a <u>useless</u> recommender system!

Outlines

1. Introduction
   - Field of research
   - Goal
2. Problem of Serendipity
   - State-of-the-art
   - Formalization of the problem
   - Method
3. Results
   - Datasets
   - Setting
   - Quantitative Results
   - Qualitative Results
4. Conclusion

1. Introduction
   - Field of research
   - Goal
2. Problem of Serendipity
   - State-of-the-art
   - Formalization of the problem
   - Method
3. Results
   - Datasets
   - Setting
   - Quantitative Results
   - Qualitative Results
4. Conclusion

Introduction Field of research

Accuracy $\neq$ Usefulness!
[Abbassi et al., 2009, Kunaver and Požrl, 2017], ...

Regular recommender problem

Input:

- A user u
- A set of objects V in which the recommended item must belong
- Access to the histories of the user(s): $\{(\text{object}_k, \text{reward}_k)\}_k$

Goal:

Return a recommended item
that maximizes the reward for the seller
(price, probability of buying, ...)

Introduction Field of research

Accuracy $\neq$ Usefulness!
[Abbassi et al., 2009, Kunaver and Požrl, 2017], ...

Recommender problem **with serendipity**

Input:

- A user u
- A set of objects V in which the recommended item must belong
- Access to the histories of the user(s): $\{(\text{object}_k, \text{reward}_k)\}_k$

Goal:

Return a recommended item
that maximizes **both the reward**
and the novelness.

Introduction Field of research

$\sim$ *diversity-accuracy dilemma* [Zhou et al., 2010]
$\rightarrow$ *exploration-exploitation dilemma* in bandits

Bandits are popular tools to tackle the recommender problem [Koutrika, 2018, Mary et al., 2015, Guillou et al., 2016].

Multi-Armed Bandit

1: Initialize scores associated with each action (*eg* movie)
2: Repeat
    3: Compute the score of each action
    4: Select the arm/action which maximizes the score
    5: Receive the reward and improve the computation of the score
6: **return** the arm associated with the highest score

Introduction Objectives of this project

Goals

1. Formalize the problem of **recommendation with serendipity**

Introduction Objectives of this project

Goals

1. Formalize the problem of **recommendation with serendipity**
2. Find a method to tackle this problem

Introduction Objectives of this project

Goals

1. Formalize the problem of **recommendation with serendipity**
2. Find a method to tackle this problem
3. Compare it with other bandit methods

1 Introduction
- Field of research
- Goal

2 Problem of Serendipity
- State-of-the-art
- Formalization of the problem
- Method

3 Results
- Datasets
- Setting
- Quantitative Results
- Qualitative Results

4 Conclusion

State-of-the-art

Several definitions of *serendipity*

[Abbassi et al., 2009, Murakami et al., 2007, Iaquinta et al., 2008, Kotkov et al., 2016]

State-of-the-art

Several definitions of *serendipity*
[Abbassi et al., 2009, Murakami et al., 2007, Iaquinta et al., 2008, Kotkov et al., 2016]

What one would need:

- A flexible definition
- Easy to understand and grasp
- Should fit as much as possible the concept of serendipity

Formalization

- $G(V, E)$ unweighted, undirected object similarity graph
- Histories of the user $\{(\text{object}_k, \text{reward}_k)\}_k$

$$f_u^{(k)} = \text{explored objects up to time } k$$
$$r_u^{(k)} = \text{associated reward received up to time } k$$

(*random variables*)

Serendipity value

serendipity value of an unexplored object v (of *normalized* reward variable $\tilde{r}_{v,u}^{(k)}$) at time $k > 0$ with respect to user u

$$s(v, u, k) = \mathbb{E}_{(f_u^{(k)}, \tilde{r}_u^{(k)})}[\tilde{r}_{v,u}^{(k)} \times d_e(v, \text{explored}) | (f_u^{(t)}, \tilde{r}_u^{(t)})_{t<k}]$$

Formalization

Serendipity value

serendipity value of an unexplored object v (of *normalized* reward variable $\tilde{r}_{v,u}^{(k)}$) at time $k > 0$ with respect to user u

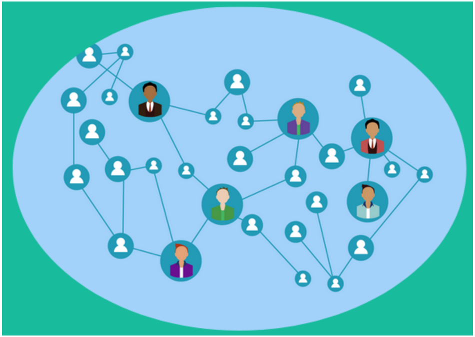
$$s(v, u, k) = \mathbb{E}_{(f_u^{(k)}, \tilde{r}_u^{(k)})}[\tilde{r}_{v,u}^{(k)} \times d_e(v, \text{explored}) | (f_u^{(t)}, \tilde{r}_u^{(t)})_{t<k}]$$

Thus the set of potential serendipities at time $k > 0$ for user u is denoted S_u

Potential Serendipities

$$S_u = \arg\max \{v \text{ unexplored} : s(v, u, k)\}$$

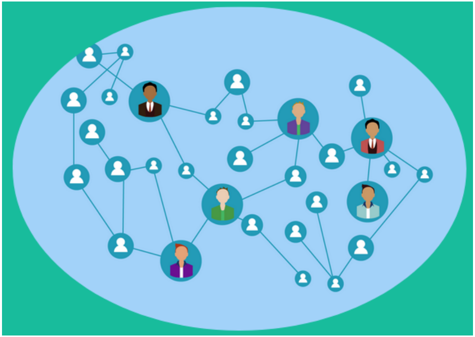
Method Adapting from Influence Maximization



richardkim.me/
influencemaximization

To whom should products be given in order to become viral?

Method Adapting from Influence Maximization

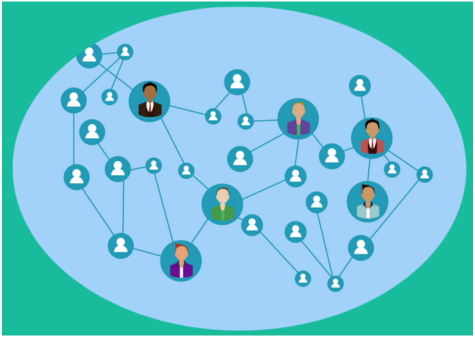


richardkim.me/
influencemaximization

To whom should products be given in order to become viral?

Online: Learning while running the marketing campaign

Method Adapting from Influence Maximization



`richardkim.me/influencemaximization`

To whom should products be given in order to become viral?

Online: Learning while running the marketing campaign

Persistent: Once a node is explored, it does not yield a reward anymore

Method Algorithm from [Lagrée et al., 2017]

Score:

$$b_k(t) = \hat{R}_k(t) + \left(1+\sqrt{2}\right)\sqrt{\frac{\hat{\lambda}_k(t)\log(4t)}{n_k(t)}} + \frac{\log(4t)}{3n_k(t)}$$

Require: Set of candidates $[K]$, time budget N

1: **Initialization:** play each candidate $k \in [K]$ once, observe the spread $S_{k,1}$, set $n_k = 1$
2: For each $k \in [K]$: update the reward $W = W \cup S_{k,1}$
3: **for** $t = K+1, \ldots, N$ **do**
4: Compute $b_k(t)$ for every candidate k
5: Choose $k(t) = \arg\max_{k \in [K]} b_k(t)$
6: Play candidate $k(t)$ and observe spread $S(t)$
7: Update cumulative reward: $W = W \cup S(t)$
8: Update statistics of candidate $k(t)$: $n_{k(t)}(t+1) = n_{k(t)}(t) + 1$ and $S_{k,n_k(t)} = S(t)$.
9: **end for**
10: **return** W

Method Adapted algorithm

Apply the serendipity constraint on the set of candidates (and their supports) selected in [Lagrée et al., 2017]'s algorithm, parametrized by value s.

W is the adjacency matrix of the object graph, F feature matrix.

$S \leftarrow \text{Supp}(f_u^{(t)})^c \cap \{c \in V : \exists i, 1 \leq i \leq s, W^i[c, \text{Supp}(f_u^{(t)})]\mathbf{1} > 0\}$
centroids $\leftarrow$ Kmeans(data $= F[S, :]$, nclusters $= K$)
candidates $\leftarrow \emptyset$
for $c \in$ centroids **do**
  Append $\arg\min_{v \in S} ||F[v, :] - F[c, :]||_2^2$ to candidates
end for
supports $\leftarrow \emptyset$
for $v \in$ candidates **do**
  Append $\{v' \in S : W[v, v'] > 0\}$ to supports
end for
return candidates, supports

1 Introduction
- Field of research
- Goal

2 Problem of Serendipity
- State-of-the-art
- Formalization of the problem
- Method

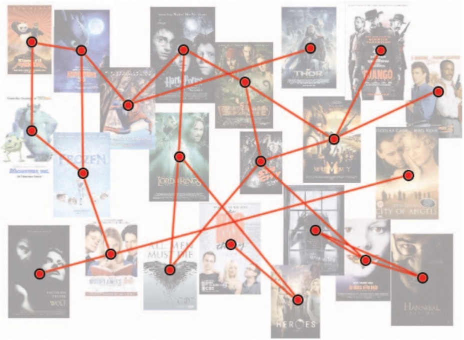
3 Results
- Datasets
- Setting
- Quantitative Results
- Qualitative Results

4 Conclusion

MovieLens

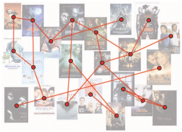
Movie recommendation! data on users, movies, and ratings

(*GroupLens Research*: `movielens.org`)



M. Valko's slides for Lecture 7

MovieLens (ml-1m, ml-20m)



- **MovieLens 1M Dataset (ml-1m)**

#movies	#users	average #ratings/user
4,000	6,000	165

- **MovieLens 20M Dataset (ml-20m)**

#movies	#users	average #ratings/user
27,000	138,000	144

Benchmark on *MovieLens* (ml-1m, ml-20m)

Evaluation on 100 iterations and at horizon 100

Cumulative regret

$a^{(t)}$ is the recommended item at time t:

$$R_T = \sum_{t \leq T} \max\{a^* \text{unexplored} : r(a^*)\} - r(a^{(t)})$$

Benchmark on *MovieLens* (ml-1m, ml-20m)

Evaluation on 100 iterations and at horizon 100

Cumulative regret

$a^{(t)}$ is the recommended item at time t:

$$R_T = \sum_{t \leq T} \text{max}\{a^* \text{unexplored} : r(a^*)\} - r(a^{(t)})$$

Diversity measure ([Vie, 2016])

k is the number of rounds, $V^{(k)}$ is the feature matrix of explored objects up to time k

$$D(V^{(k)}) = \sqrt{|V^{(k)}.t(V^{(k)})|}$$

Benchmark on *MovieLens* (ml-1m, ml-20m)

Algorithms

Tested bandit methods

1. Random strategy

Benchmark on *MovieLens* (ml-1m, ml-20m)

Algorithms

Tested bandit methods

1. Random strategy
2. ϵ-greedy strategy (w.r.t. diversity measure)

Benchmark on *MovieLens* (ml-1m, ml-20m)

Algorithms

Tested bandit methods

1. Random strategy
2. ϵ-greedy strategy (w.r.t. diversity measure)
3. LinUCB (described in [Chu et al., 2011])

Benchmark on *MovieLens* (ml-1m, ml-20m)

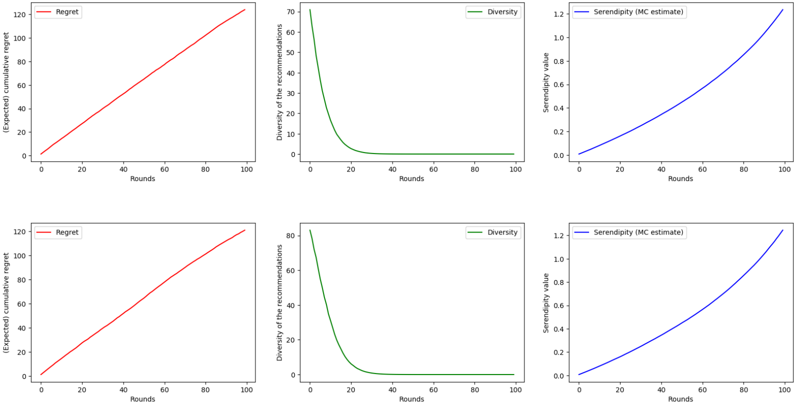
Algorithms

Tested bandit methods

1. Random strategy
2. ϵ-greedy strategy (w.r.t. diversity measure)
3. LinUCB (described in [Chu et al., 2011])
4. The adapted method from [Lagrée et al., 2017]

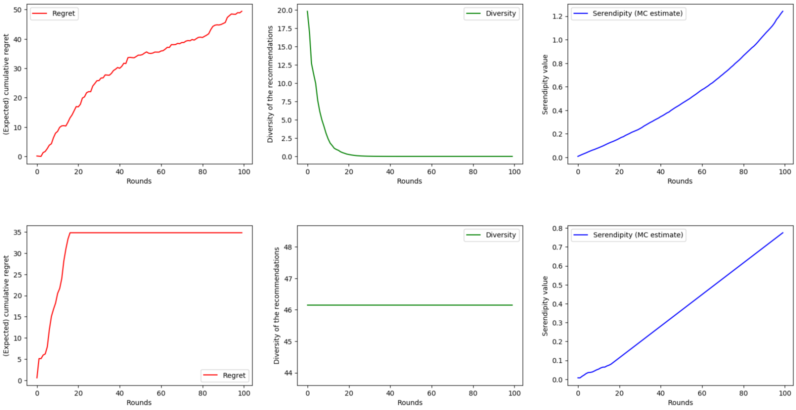
Results Regret, diversity, serendipity curves

ml-20m dataset, random (*top*) and ϵ-greedy methods

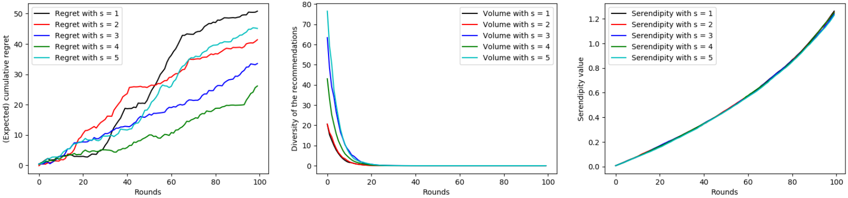


Results Regret, diversity, serendipity curves

ml-20m dataset, adapted method (*top*) and LinUCB methods



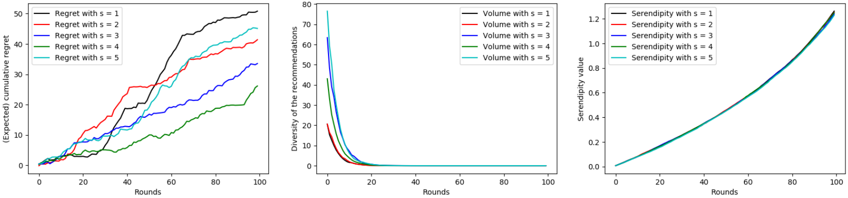
Results Variation of parameter s



Remarks:

- When s increases (up to some point), diversity increases and regret decreases

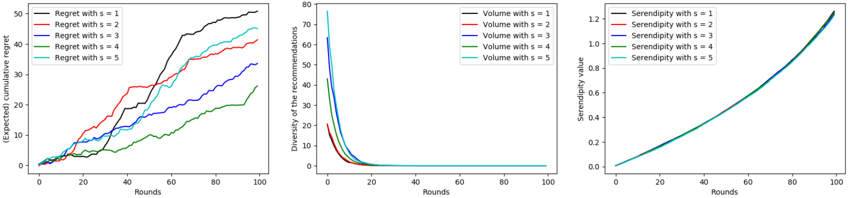
Results Variation of parameter s



Remarks:

- When s increases (up to some point), diversity increases and regret decreases
- From some value of s, regret increases again

Results Variation of parameter s



Remarks:

- When s increases (up to some point), diversity increases and regret decreases
- From some value of s, regret increases again
- Little influence of s on the (cumulative) serendipity value...

1 Introduction
- Field of research
- Goal

2 Problem of Serendipity
- State-of-the-art
- Formalization of the problem
- Method

3 Results
- Datasets
- Setting
- Quantitative Results
- Qualitative Results

4 Conclusion

Outlook

1. A measure for serendipity

Outlook

1. A measure for serendipity
2. A method that tackles the problem of recommendation with serendipity

Outlook

1. A measure for serendipity
2. A method that tackles the problem of recommendation with serendipity
3. Benchmark with classic/naive methods of recommendation

Outlook

1. A measure for serendipity
2. A method that tackles the problem of recommendation with serendipity
3. Benchmark with classic/naive methods of recommendation

What remains to be done

- Testing with methods that might be more relevant: Rotting Bandits [Seznec et al., 2018], Outside-The-Box recommendation [Abbassi et al., 2009]

Outlook

1. A measure for serendipity
2. A method that tackles the problem of recommendation with serendipity
3. Benchmark with classic/naive methods of recommendation

What remains to be done

- Testing with methods that might be more relevant: Rotting Bandits [Seznec et al., 2018], Outside-The-Box recommendation [Abbassi et al., 2009]
- User similarity has been ignored here

Outlook

1. A measure for serendipity
2. A method that tackles the problem of recommendation with serendipity
3. Benchmark with classic/naive methods of recommendation

What remains to be done

- Testing with methods that might be more relevant: Rotting Bandits [Seznec et al., 2018], Outside-The-Box recommendation [Abbassi et al., 2009]
- User similarity has been ignored here
- The parametrization with the serendipity threshold s has an influence on regret and diversity, but not so much on the serendipity measure!!

References I

Abbassi, Z., Amer-Yahia, S., Lakshmanan, L. V., Vassilvitskii, S., and Yu, C. (2009).
Getting recommender systems to think outside the box.
In *Proceedings of the third ACM conference on Recommender systems*, pages 285–288. ACM.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011).
Contextual bandits with linear payoff functions.
In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214.

Guillou, F., Gaudel, R., and Preux, P. (2016).
Large-scale bandit recommender system.
In *International Workshop on Machine Learning, Optimization and Big Data*, pages 204–215. Springer.

References II

Iaquinta, L., De Gemmis, M., Lops, P., Semeraro, G., Filannino, M., and Molino, P. (2008).
Introducing serendipity in a content-based recommender system.

In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, pages 168–173. IEEE.

Kotkov, D., Wang, S., and Veijalainen, J. (2016).
A survey of serendipity in recommender systems.
Knowledge-Based Systems, 111:180–192.

Koutrika, G. (2018).
Recent advances in recommender systems: Matrices, bandits, and blenders.

References III

Kunaver, M. and Požrl, T. (2017).
Diversity in recommender systems–a survey.
Knowledge-Based Systems, 123:154–162.

Lagrée, P., Cappé, O., Cautis, B., and Maniu, S. (2017).
Effective large-scale online influence maximization.
In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 937–942. IEEE.

Mary, J., Gaudel, R., and Preux, P. (2015).
Bandits and recommender systems.
In *International Workshop on Machine Learning, Optimization and Big Data*, pages 325–336. Springer.

References IV

Murakami, T., Mori, K., and Orihara, R. (2007).
Metrics for evaluating the serendipity of recommendation lists.
In *Annual conference of the Japanese society for artificial intelligence*, pages 40–46. Springer.

Seznec, J., Locatelli, A., Carpentier, A., Lazaric, A., and Valko, M. (2018).
Rotting bandits are no harder than stochastic ones.
arXiv preprint arXiv:1811.11043.

Vie, J.-J. (2016).
Modèles de tests adaptatifs pour le diagnostic de connaissances dans un cadre d'apprentissage à grande échelle.
PhD thesis, Université Paris-Saclay.

References V

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. (2010).
Solving the apparent diversity-accuracy dilemma of recommender systems.
Proceedings of the National Academy of Sciences, 107(10):4511–4515.