Improving Relevance in a Recommendation System to Suggest Charities

Pablo Adames, U. of Calgary, now with CP Railway IS
Sourabh Mokhasi, U. of Calgary, now with AWS

Data for Good, April 28, 2022

Table of Contents

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for...

- An user-based collaborative filter algorithm

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person ✎
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for...

- An user-based collaborative filter algorithm

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for...

- An user-based collaborative filter algorithm

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for...

- An user-based collaborative filter algorithm

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for...

- An user-based collaborative filter algorithm

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for... ✎

- An user-based collaborative filter algorithm

Everyday recommendations 1

Recommending birthday gifts...:

- Given a description or knowledge of a person
- you come up with gift ideas
- based on preferences or choices of other people you know
- that have similar taste or preferences as the birthday person

This is the basis for...

- An user-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

- An item-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

- An item-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas ✎
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

- An item-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

- An item-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

- An item-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

- An item-based collaborative filter algorithm

Everyday recommendations 2

Recommending birthday gifts...:

- Given a description or knowledge of what a person prefers or uses
- you come up with gift ideas
- that are like things you have seen other people prefer
- who have chosen the same items as the birthday person

This is the basis for ...

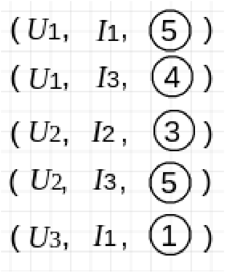
- An item-based collaborative filter algorithm

Recommendation system algorithms - data

1. Recommendations can be calculated with algorithms
2. They expect data as a list of $(user, item, rating)$
3. Or equivalently as a rating matrix of user vs. items

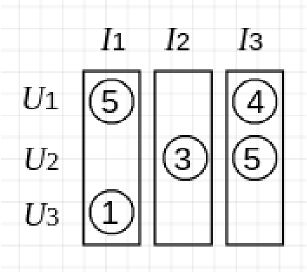
Recommendation system algorithms - data

1. Recommendations can be calculated with algorithms
2. They expect data as a list of $(user, item, rating)$
3. Or equivalently as a rating matrix of user vs. items



Recommendation system algorithms - data

1. Recommendations can be calculated with algorithms
2. They expect data as a list of (*user*, *item*, *rating*)
3. Or equivalently as a rating matrix of user vs. items



Recommendation system algorithms - tasks

1. Recommendation algorithms implement these main tasks
2. A definition of similarity to create neighbourhoods
3. Rating prediction
4. To produce a ranked list of items for the target user

Recommendation system algorithms - tasks

1. Recommendation algorithms implement these main tasks
2. A definition of similarity to create neighbourhoods
3. Rating prediction
4. To produce a ranked list of items for the target user

- Item-based CF: Similar items to the ones rated by the target user
- User-based CF: Similar users that have rated the same items as the target user

Recommendation system algorithms - tasks

1. Recommendation algorithms implement these main tasks
2. A definition of similarity to create neighbourhoods
3. Rating prediction
4. To produce a ranked list of items for the target user

- Item-based CF: User-similarity weighted ratings to unseen item
- User-based CF: Average target user rating corrected by user-similarity weighted rating to unseen item

Recommendation system algorithms - tasks

1. Recommendation algorithms implement these main tasks
2. A definition of similarity to create neighbourhoods
3. Rating prediction
4. To produce a ranked list of items for the target user

- Descending order from top predicted rated item and length k
- Top-k recommendations

Algorithm implementations - elements

1. Naive implementations work on the user-item matrix
2. Sparse matrix puts high tax on memory
3. Algorithm complexity puts high tax on processing
4. SVD++ and auto-encoders tackle sparsity and complexity...
5. Real applications store a lot of pre-computed values

Algorithm implementations - elements

1. Naive implementations work on the user-item matrix
2. Sparse matrix puts high tax on memory
3. Algorithm complexity puts high tax on processing
4. SVD++ and auto-encoders tackle sparsity and complexity...
5. Real applications store a lot of pre-computed values

Algorithm implementations - elements

1. Naive implementations work on the user-item matrix
2. Sparse matrix puts high tax on memory
3. Algorithm complexity puts high tax on processing
4. SVD++ and auto-encoders tackle sparsity and complexity...
5. Real applications store a lot of pre-computed values

Algorithm implementations - elements

1. Naive implementations work on the user-item matrix
2. Sparse matrix puts high tax on memory
3. Algorithm complexity puts high tax on processing
4. SVD++ and auto-encoders tackle sparsity and complexity...
5. Real applications store a lot of pre-computed values

Algorithm implementations - elements

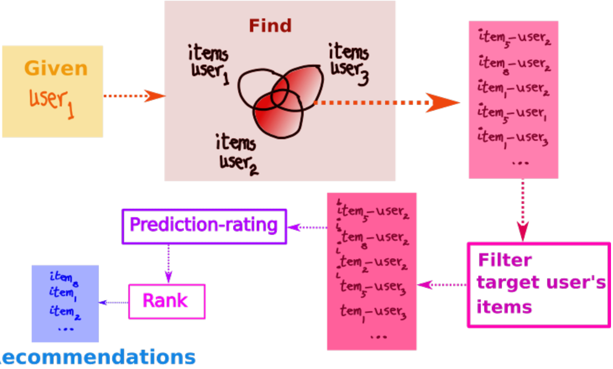
1. Naive implementations work on the user-item matrix
2. Sparse matrix puts high tax on memory
3. Algorithm complexity puts high tax on processing

4. SVD++ and auto-encoders tackle sparsity and complexity...
5. Real applications store a lot of pre-computed values

CF algorithm with dual-autoencoders

The recommendation algorithm

A collaborative filter based on item similarity



CF algorithm with dual-autoencoders

The ML recommendation components

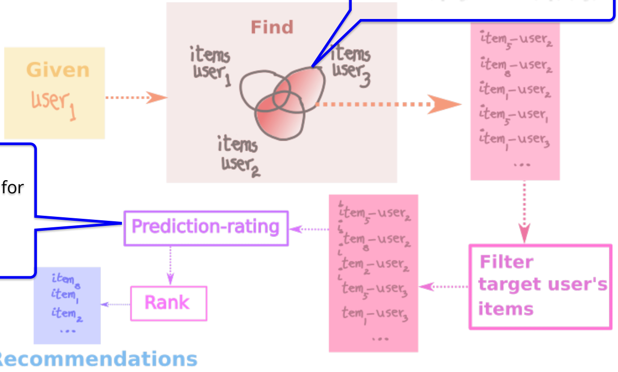
We need these components to compute recommendations

Similar causes via distance between cause-embedding vectors:

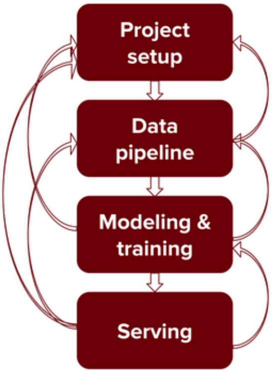
$$sim(c_1, c_3) = 1.0 - |c_1| \cdot |c_3|$$

Prediction of implicit rating for unseen causes

$$f(u_1, c_5) \rightarrow r_{1,5}$$



The Machine Learning Project Flow



From https://huyenchip.com/machine-learning-systems-design/design-a-machine-learning-system.htmldesign-a-machine-learning-system-dwGQI5R

Table of Contents

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

Context

Given the following:

- A global charitable donations platform ✎
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

meetup

data for good

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data ✎
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories ✎
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF) ✎

Build in the cloud

- A prototype to make recommendations

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

Context

Given the following:

- A global charitable donations platform
- Real-life anonymous transaction data
- No explicit user profiles
- Causes with categories and subcategories
- Collaborative filter (CF)

Build in the cloud

- A prototype to make recommendations

Donations related previous work

- 59,000 ratings of 70 non-profit organizations from over 3,800 users, May 2009
- On-boarding interview to obtain user donation preferences
- Patented constant-time algorithm for recommendations: EigenTaste 2.0

Algorithm selection

Options for the basic recommendation algorithm:

- Content-based
- Item-based collaborative filter
- User-based collaborative filter

Settled for Item-based CF due to:

- CF favours personalization and serendipity
- Lack of user profiles

ML selection

Strategies we considered:

- Dual-autoencoders
- Scalable SVD

Settled for dual-autoencoders due to:

- Efficient use of memory
- Parallel pipeline design

System metric selection

Could use:

- MAP
- Diversity
- Serendipity
- Novelty
- Coverage

Chose MAP for simplicity

See K. Falk, Practical Recommender Systems. Shelter Island, NY: Manning Publications, 2019 Kaminskas, Marius and Bridge, and Derek, "Diversity, Serendipity, Novelty, and Coverage", ACM transactions on interactive intelligent systems, v.7, No. 1, pp. 1-14, 2016

Question

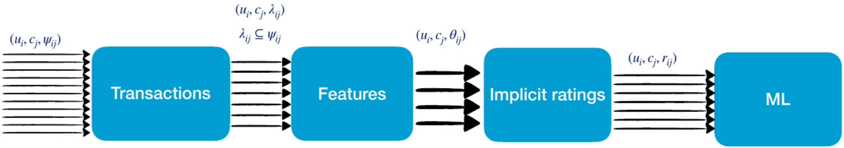
Given:

- Lack of explicit user profiles or ratings
- Sparsity:
  - 24 million anonymous donations
  - 165 thousand unique causes
  - Over 1.2 million users

How to make relevant recommendations?

Table of Contents

Generation of features



To capture patterns from how users interact with causes

- The set of transactions ψ gets pre-processed into λ
- Then the set of features θ is computed
- Finally the set of implicit ratings, r, is generated

Features

FOCUS: how a user distributes their donation money among causes

INTENSITY: capture the frequency of donations by a user to a category of causes

IMPACT: It is very similar to INTENSITY but in terms of money

KIND: encodes the category of the cause donated to

Data set description

Table: Statistics of data sets to make recommendations for donations to charities.

Dataset	No. of users	No. of items	No. of ratings	Rating density	Avg. ratings per user	Avg. ratings per item
Donations-Dashboard[1]	3 133	70	57 517	26.31%	18.42	851.9
Donations-1[2]	9 397	2 711	24 417	0.0958%	2.60	9.03
Donations-2[2]	1 264 083	165 842	24 174 672	0.0115%	19.12	145.76

[1] The Donation Dashboard data set: http://dd.berkeley.edu/dataset/

[2] Data used for this research.

Note the sparsity in rating density[a] of the Donations-2 data set.

[a] The ratings per user per item

ML metrics

Measuring what matters: ☺

For the machine learning components...

Regressor performance: RMSE, MAE ✓

Classifier performance: precision, recall ✓

Used during training and tuning via cross-validation ✓

The goal is to generalize while avoiding over-fitting ✓

ML metrics

Measuring what matters:

For the machine learning components...

Regressor performance: RMSE, MAE ✓

Classifier performance: precision, recall ✓

Used during training and tuning via cross-validation ✓

The goal is to generalize while avoiding over-fitting ✓

ML metrics

Measuring what matters:

For the machine learning components...

Regressor performance: RMSE, MAE ✓

Classifier performance: precision, recall ✓

Used during training and tuning via cross-validation ✓

The goal is to generalize while avoiding over-fitting ✓

ML metrics

Measuring what matters:

For the machine learning components...

Regressor performance: RMSE, MAE ✓

Classifier performance: precision, recall ✓

Used during training and tuning via cross-validation ✓

The goal is to generalize while avoiding over-fitting ✓

ML metrics

Measuring what matters:

For the machine learning components...
Regressor performance: RMSE, MAE ✓
Classifier performance: precision, recall ✓
Used during training and tuning via cross-validation ✓
The goal is to generalize while avoiding over-fitting ✓

ML metrics

Measuring what matters:

For the machine learning components...

Regressor performance: RMSE, MAE ✓

Classifier performance: precision, recall ✓

Used during training and tuning via cross-validation ✓

The goal is to generalize while avoiding over-fitting ✓

System performance metrics

System Performance:
A list of recommendations to a user...
is a ranked list of values
and MAP is a more suitable metric.
It requires that we can determine for each element
of the list if it is relevant to the user

System performance metrics

System Performance:

A list of recommendations to a user...
is a ranked list of values
and MAP is a more suitable metric.
It requires that we can determine for each element
of the list if it is relevant to the user

System performance metrics

System Performance:

A list of recommendations to a user...

is a ranked list of values

and MAP is a more suitable metric.

It requires that we can determine for each element of the list if it is relevant to the user

System performance metrics

System Performance:

A list of recommendations to a user...
is a ranked list of values
and MAP is a more suitable metric.
It requires that we can determine for each element of the list if it is relevant to the user

System performance metrics

System Performance:
A list of recommendations to a user...
is a ranked list of values
and MAP is a more suitable metric.
It requires that we can determine for each element of the list if it is relevant to the user

System performance metrics

System Performance:
A list of recommendations to a user...
is a ranked list of values
and MAP is a more suitable metric.
It requires that we can determine for each element
of the list if it is relevant to the user

Table of Contents

Sensitivity to embedding size

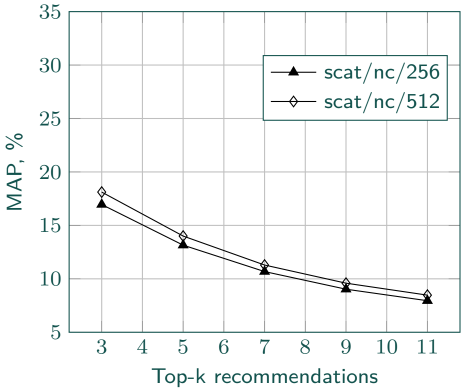


Figure: ***Effect of the size of the cause embedding vectors on MAP at various values of top k for samples of 10,000 users. Key: scat=subcategories of causes used to compute features, nc=no feature for country of user, 256=embedding size, 512=embedding size.***

Cause categories Vs. subcategories

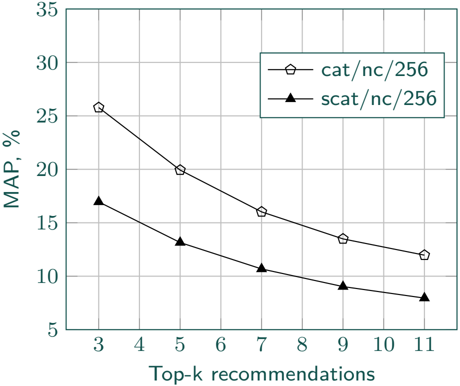


Figure: ***Effect of using categories or subcategories on MAP at different top k values for samples of 10,000 users. Key: cat=categories for feature computation, scat= subcategories instead of categories for feature computation, nc=no feature for country of user, 256=size of the cause embedding vectors.***

Effect of user activity

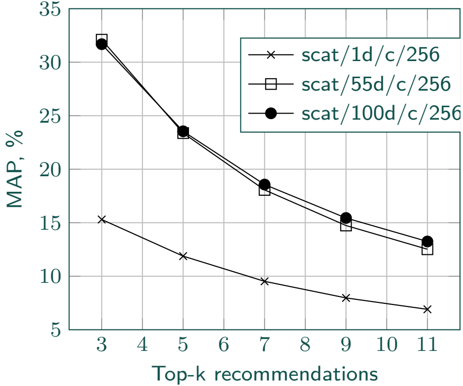


Figure: ***Effect of removing users with a minimum number of donations on MAP at various top k values for samples of 10,000 users. Key: scat= subcategories used for feature computation, c= country of the user as a feature, 1d=all users are included in the system, 55d=only users with 55 donations or more are included, 100d=only users with more than 100 donations are included.***

Transactions and users

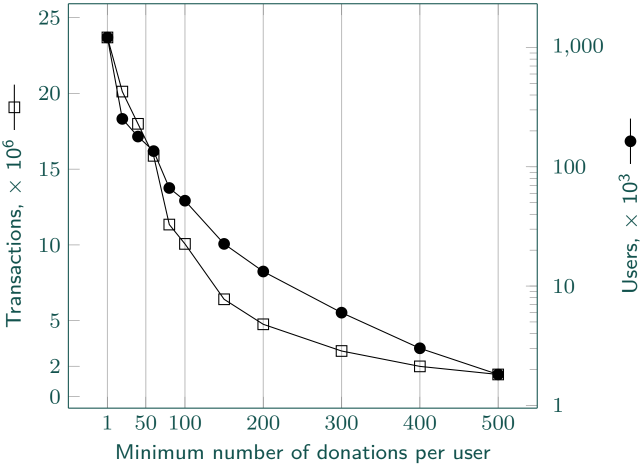


Figure: ***Number of transactions and number of users in the data set as a function of the minimum number of donations per user during the 2-year interval in the donations-2 data set.***

Causes and companies

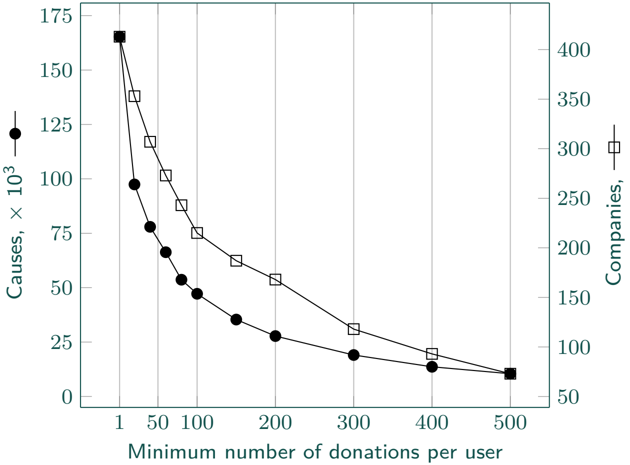


Figure: ***Number of causes and companies in the data set as a function of the minimum number of donations per user during the 2-year interval in the donations-2 data set.***

Countries of users

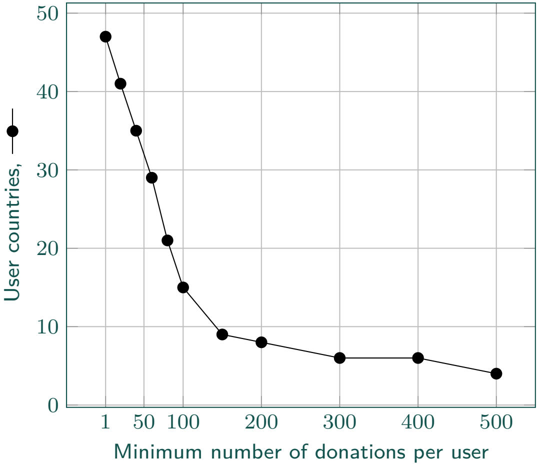


Figure: ***Number of user countries in the data set as a function of the minimum number of donations per user during the 2-year interval in the donations-2 data set.***

Table of Contents

Conclusions

- Successfully used inferred user profiles via implicit ratings
- Using finer cause classifications increases relevance
- Controlling for minimum number of donations improves MAP
- An empirical threshold for user donations seems reasonable to max MAP at any k

Table of Contents

Acknowledgements

Special thanks for funding and supporting this research:

- The University of Calgary, Software Engineering Program
- MITACS Accelerate
- Benevity, Inc.