

Linked Data を用いたユーザの興味に基づく映画推薦手法の提案

上松 陽介[†] 疋田 輝雄[†]

明治大学理工学研究科[†]

1. はじめに

近年, Linked Data のデータ量や種類が増加し, その様々な利用方法がある. たとえば英国の BBC 放送の番組情報ページの作成や DBpedia Mobile での地図へのアノテーションがあげられる. 我々は, Linked Data に, ユーザから取得した個人の興味に関する値とリンク解析アルゴリズム HITS を用いることで, 従来の推薦手法よりもユーザの興味に沿った情報の推薦が可能になると考える. 我々の提案手法は, ユーザの興味に合ったリソースに関わる作品ほど優先的に推薦する特徴を持つ.

本稿では, 我々の提案手法と評価のために作成したプロトタイプ映画推薦システムについて報告する. プロトタイプは, Linked Data の一種である DBpedia^{*1}を対象に実装を行った.

2. 関連研究との比較

Linked Data を用いた推薦システムには, 一瀬たちや R.Mirizzi たちの研究がある. 一瀬たち[1]は, リソースに関係するリソースの数に着目した大西たちが提案したスコアを用いて, 作家を推薦する手法を提案した. 本研究は, 各プロパティに異なる重みを与える点で異なり, ユーザの興味に合わせた推薦を行うことができる特徴を持つ.

R.Mirizzi たち[2]は, TFIDF とコサイン類似度を用いて映画同士を比べた類似度の計算を行い, ユーザが気に入っている映画と似ている映画を推薦する手法を提案した. R.Mirizzi らの研究は, TFIDF を用いているため希少なリソースに対して高いスコアを付ける傾向がある. 本研究は, HITS を用いているため一般的なリンクに対して高いスコアを付ける傾向がある点で異なり, 一般的なリソースと関わるリソースを優先して推薦する.

3. 提案手法

本手法では, まず対象リソースと, ユーザが抱いている興味に合わせた値をユーザから取得する. 次に, 取得したリソースに関係するリソースを DBpedia から取得し, リソースの距離やプロパティに着目するように改良した HITS を用いて推薦を行う.

3.1 リソースの取得とグラフの作成

ユーザから取得した「対象リソース」(たとえば映画)をもとに, DBpedia から「関連リソース」(たとえば監督や俳優)と「推薦候補リソース」(映画)を取得する. ここで, 関連リソースとは, 対象リソースまたは推薦候補リソースに対し推薦する際に重要な要素としたプロパティ集合Pでつながっているリソースである. また, 推薦候補リソースとは, 対象リソースに対し同じ関連リソースでつながっているリソースである.

関連リソースと推薦候補リソースを取得後, これらよりグラフを作成する. 作成するグラフは, 取得した各リソースをノードとし, 各ノード間をつなぐプロパティをエッジとした有向グラフである. エッジは, 関連リソースのノードから他ノードへの向きとノード間のプロパティを, 各ノードは対象リソースまでに経由するノード数である距離dを保持する. ここでは, 距離3までのリソースでグラフを構成する.

3.2 推薦リソースの計算

作成したグラフに対し, ユーザの興味とリソースの距離を考慮するように HITS を改良した手順 1-4 を用いてスコアを算出する. なお, HITS は各ページに対して Authority Score と Hub Score を定義し, あるテーマにおける優良な参照関係を見つけ出し, 重要性の高い Web ページを抽出するアルゴリズムである. Authority Score とは,

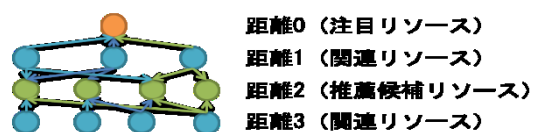


図1 対象リソースの有向グラフ

Proposal for Movie Recommendation Based on User Interest by Using Linked Data

†Yosuke Uematsu and Teruo Hikita

School of Science and Technology, Meiji University

*¹<http://dbpedia.org/About/>

重要な情報を発信しているかを示すスコアである。Hub Score とは、重要な情報を発信している Web ページに発リンクしているかを示すスコアである。

手順 1 Hub Score の初期化

各ノードの Hub Score に初期値として 1.0 を与える。

手順 2 Authority Score の計算

各ノード*i*の Authority Score (as_i) は、自身に矢印を向けているノード群*E*から求める。 as_i は、隣接ノード*j*∈*E*の Hub Score (hs_j) と、ノード*j*の間のプロパティ*p*に対するユーザの興味を示す値 α_{ip} の積を、ノード*j*の距離 d_{ij} を 2 乗した値で割った値の総和になる。この計算により、対象リソースに関係が強いより興味の高い関係のリソースとつながっているノードのスコアが高くなる。

$$as_i = \sum_{j \in E} \sum_{p \in P} \frac{\alpha_{ip} * hs_j}{(d_{ij} + 1)^2}$$

手順 3 Hub Score の計算

各ノード*i*の Hub Score (hs_i) は、自身が矢印を向けているノード群*E*から求める。 hs_i は、隣接ノード*j*∈*E*の Authority Score (as_j) を、ノード*j*の距離 d_{ij} を 2 乗した値で割った値の総和になる。この計算により、対象リソースに関係が強いより有名なノードのスコアが高くなる。

$$hs_i = \sum_{j \in E} \frac{as_j}{(d_{ij} + 1)^2}$$

手順 4 各スコアの正規化

各ノードの Hub Score と Authority Score を、全ノードの該当スコアの 2 乗和の平方根で割り正規化する。各ノードのスコアが収束していなければ手順 2 へ戻る。

計算終了後の Authority Score が高いリソースは、対象リソースをもとにユーザの興味に沿った重要なリソースになり、スコアが高い順に推薦リソースとしてユーザに提示される。

4. 実験例

提案手法を適用した映画推薦システムを実装し実験を行った。このシステムでは、重要な要素であるプロパティ群*P*を*P* = {俳優, 脚本家, 監督} とした。映画「THE 有頂天ホテル」(候補リソース数は 70, 関連リソース数は 302) を対象に 2 通りの趣向のユーザに対して推薦を行った。

A. 「映画は俳優で決める。監督・脚本家はほとんど気にしない」 (*P*={10, 2, 2})

表 1 ユーザ A に対する推薦結果

順位	推薦リソース (映画名)	スコア
1	叫	0.15090
2	回路 (映画)	0.14688
3	アンフェア the answer	0.13848
4	ステキな金縛り	0.09622
5	赤い橋の下のぬるい水	0.07806

表 2 ユーザ B に対する推薦結果

順位	推薦リソース (映画名)	スコア
1	ステキな金縛り	0.37459
2	みんなのいえ	0.34465
3	ラヂオの時間	0.34384
4	ザ・マジックアワー	0.34260
5	叫	0.06369

推薦結果を表 1 に示す。結果を見ると、1-3 位には、対象の映画と共通する俳優が 2 人ずつ出演している映画が来ている。また、4-5 位には、対象の映画の出演俳優の中で最も出演作品が多い「役所広司」が出演している映画が来ている。これらの結果より、俳優に対する関係が強い映画が推薦されていることが分かる。

B. 「映画は監督・脚本家で決める。俳優はほとんど気にしない」 (*P*={2, 10, 10})

推薦結果を表 2 に示す。結果を見ると、1-4 位には、対象の映画の監督・脚本家である「三谷幸喜」の映画が来ている。また、5 位には、三谷幸喜の作品によく出演している俳優が出演している映画が来ている。これは、三谷幸喜の関連する作品が DBpedia にこれ以外にないためである。これらの結果より、監督・脚本家に対する関係が強い映画が推薦されていることが分かる。

実験結果より、興味に合わせた推薦がされることが確認できる。他の映画（「七人の侍」等）でも興味に合わせた推薦を行うことができ、提案手法が有効であることを確認できた。

5. おわりに

本稿では、ユーザの興味に沿った情報を Linked Data を用いて推薦する手法を提案し、プロトタイプのシステムについて報告を行った。今後は、関連リソース間の関係も考慮する手法を考えていきたい。

参考文献

[1] 一瀬, 大西, 小林: DBpedia を利用した作家推薦の試み, 3C1-OS-13a-5, 人工知能学会全国大会(2012).
 [2] R.Mirizzi, T.D.Noia, A.Ragone, V.C.Ostuni and E.D.Sciascio: Movie Recommendation with DBpedia, 3rd Italian IR Workshop (2012).