Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)
IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

Movie Recommender System Using Collaborative Filtering

Meenu Gupta
Department of Computer Science & Engineering
Chandigarh University, Punjab
Gupta.meenu5@gmail.com

Aditya Thakkar
Department of Computer Science & Engineering Chandigarh University, Punjab
dadityа571@gmail.com

Aashish
Department of Computer Science & Engineering
Chandigarh University, Punjab
aashish.saini26@gmail.com

Vishal Gupta
Department of Computer Science & Engineering
Chandigarh University, Punjab
premvishal9090@gmail.com

Dhruv Pratap Singh Rathore
Department of Computer Science & Engineering
Chandigarh University, Punjab
dhruv.rathore99@gmail.com

Abstract: **Movies are one of the sources of entertainment, but the problem is in finding the desired content from the ever-increasing millions of content every year. However, recommendation systems come much handier in these situations. The aim of this paper is to improve the accuracy and performance of a regular filtering technique. Although varieties of methods are used to implement a recommendation system, Content-based filtering is the simplest method. Which takes input from the users, rechecks his/her history/past behavior, and recommends a list of similar movies. In this paper, to prove the effectiveness, K-NN algorithms and collaborative filtering are used to mainly focus on enhancing the accuracy of results as compared to content-based filtering. This approach is based on cosine similarity using k-nearest neighbor with the help of a collaborative filtering technique, at the same time removing the drawbacks of the content-based filtering. Although using Euclidean distance is preferred, cosine similarity is used as the accuracy of cosine angle and the equidistance of movies remain almost the same.**

Keywords— Movie recommender system, cosine similarity, K-NN algorithms, content-based filtering, collaborative filtering, nearest neighbors

I. INTRODUCTION

Recommendation systems are predicting systems that radically recommend items to users or users to the items, and sometimes users to users too. Tech giants like YouTube, Amazon Prime, Netflix use similar methods to recommend video content according to their desired interest. As the internet contains huge loads of data, finding your content is very difficult and can be very time consuming, thus the recommendation plays an important role in minimizing our effort. These systems are getting more popular nowadays in various areas such as in books, videos, music, movies, and other social network sites where the recommendation is used to filter out the information. It is a tool that is using the user's information to improve the suggestion result and give out the most preferred choice. User/Customer satisfaction is key for building the tool. It is beneficial for both customers and companies, as the more satisfied the customer is, the more likely he/she would want to use the system for their ease, which would ultimately make revenues for the companies. Recommendation system should always be improved as the user choice can differ from other users and if the user is not happy with the result, he/she might not use it again which is the case with our system.

Although there are a lot of algorithms, collaborative filtering is the most popular one used by the companies as it involves user's interactions more. Collaborative filtering can predict better than content-based filtering because it analyses the user's browsing history and compares with other users and then suggests results [15]. Whereas, the content-based filtering takes the user's information as an input to find similar movies and recommends them in descending order (using cosine similarity). There's another method named context-based filtering where it extracts more information from the user like mood, release date, genre, etc., to give more efficient results. Our goal in this project was to keep our system very accurate compared to other recommendation techniques while making it as simple as possible. Content-based filtering has some drawbacks and a lack of accuracy and preciseness. So the proposed system is the collaborative filtering recommendation system using nearest neighbors.

II. RELATED WORK

There are many ways of recommending movies using Content-based, Collaborative (User-item, User-user), context-based, hybrid methods, and nowadays deep learning is also used to solve this problem.

In [1], C. S. M. Wu, D. Garg, and U. Bhandary proposed a recommendation system using collaborative filtering where a user's rating is used to suggest the list. The authors have used

the Apache Mahout framework and essentially compared the performances and efficiency of user-based & item-based recommendations.

In [2],R. E. Nakhli, H. Moradi, and M. A. Sadeghi proposed the percentage view approach for recommending movies to the users, it finds relevant movies for the customer and then compares the performance with a random movie recommendation system for showing the accuracy of the project.

In [3], a content-based recommendation system is proposed by H. W. Chen, Y. L. Wu, M. K. Hor, and C. Y. Tang using neural networks. In recent years, these are top topics for the researchers to work on when they want to build a movie recommendation system.

Different terminology used in implementation of movie recommender system is discussed below.

A. Content-based Filtering

This recommendation system requires some data or information on what the user might like or what his previous watched history. It is based on previous action or explicit feedback. Most of the systems in the industry don't use this approach as they require data or they are not reliable enough. For example, if a person watches the education documentary genre more multiple times than the action genre, the person is more likely to see the most-watched genre in the descending order. The figure 1 below explains the process.

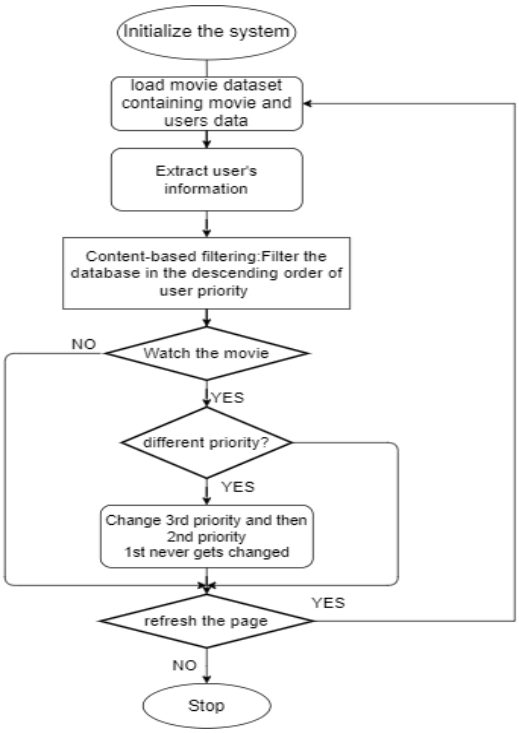

Figure 1: Content-based filtering

As in [9], R. Van Meteren, and M. Van Someren created a recommendation system by comparing the profile of the user with the content of each document in the sets of the collection. These sets of terms can be represented as the content of the document. The content-based system uses data of users and interest and browsing history to determine the results. As this requires a lot of domain knowledge, thus becomes a drawback compared to collaborative filtering.

B. Collaborative Filtering (CF)

Filters out the content according to user similar interest with other users, it basically recommends the items to users that have similar taste [13]. It is also a popular and famous algorithm in the industries. In the memory-based techniques, there are two popular filtering algorithms [10]. There is another technique known as model-based which is not as reliable as compared to memory-based techniques [17]. Figure 2 and figure 3 discussed about the item based and user based collaborative filtering.

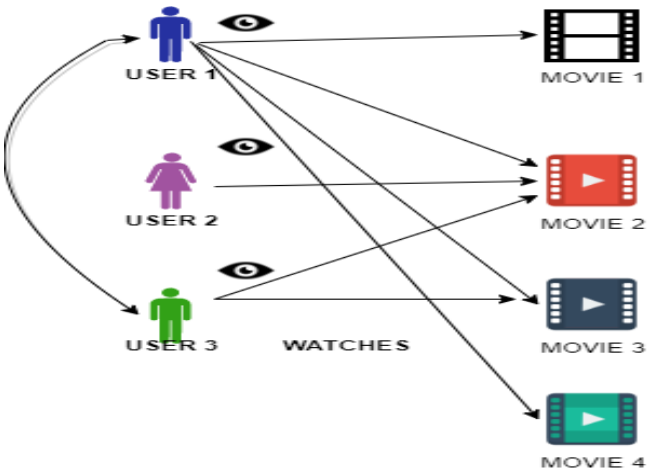

Figure 2: Demonstration of User-Based CF

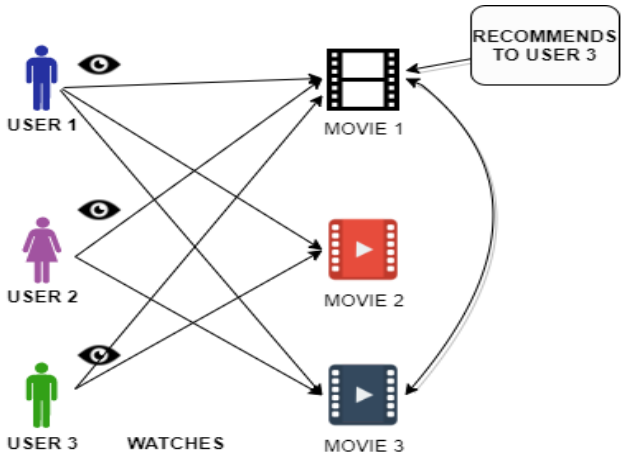

Figure 3: Demonstration of Item Based CF

In [4] the user-based, it is assumed that the user will like the items that are liked by users with whom they have similar tastes.

Consider Table 1 as an example, all the users like item A and people who like item A also like Item C, Item-based are not-dynamic in nature and do not change.

Table 1: Item-Based CF

USER/ITEM	ITEM A	ITEM B	ITEM C
USER A	✓		✓
USER B	✓	✓	✓
USER C	✓		Recommended

In the item-based like in [8], it is assumed that the user will like those items that are similar to the other items liked before. The hybrid approach-This approach provides very accurate results using both collaborative and content-based filtering while removing the drawbacks of the algorithms at the same time. This integrated system is getting more attention nowadays as it is better than both the algorithms [7].

III. PROPOSED RECOMMENDATION ENGINE

The proposed recommendation system used the collaborative filtering technique (item-based approach) which is far more accurate and more efficient to use, as the item based method can be done offline and because of its non-dynamic nature whereas the user-based changes. The proposed approach uses the KNN algorithm to find the distance between the target movies with every other movie in the dataset and then it ranks the top k nearest similar movies using cosine angle similarity. Different techniques used in this proposed algorithm are discussed below:

- **KNN algorithm-** is famous in a recommendation system for its faster predictive nature and low calculation time. KNN [16] classifies any unlabeled class to their respective classes by prediction on a similarity measure as shown in figure 4.
- **Cosine similarity-** to calculate the distance between the target movie and the movies in the dataset, cosine similarity is used. It measures the similarity between two documents irrespective of how different they are in size, and calculates the cosine angle between two vectors in multi-dimensional space[6],

$$\theta = \frac{\vec{a}.\vec{b}}{||\vec{a}||.||\vec{b}||} \qquad Eq.(1)$$

Eq. (1) is used to define the cosine similarity of the proposed model.

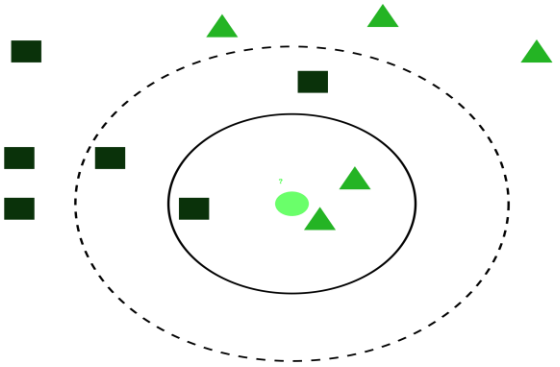

Figure 4: Demonstration of the K-NN algorithm (value of k=3) [5].

- **Item-based collaborative filtering-** assumes users will like items that are similar to the items that are liked before by the user.

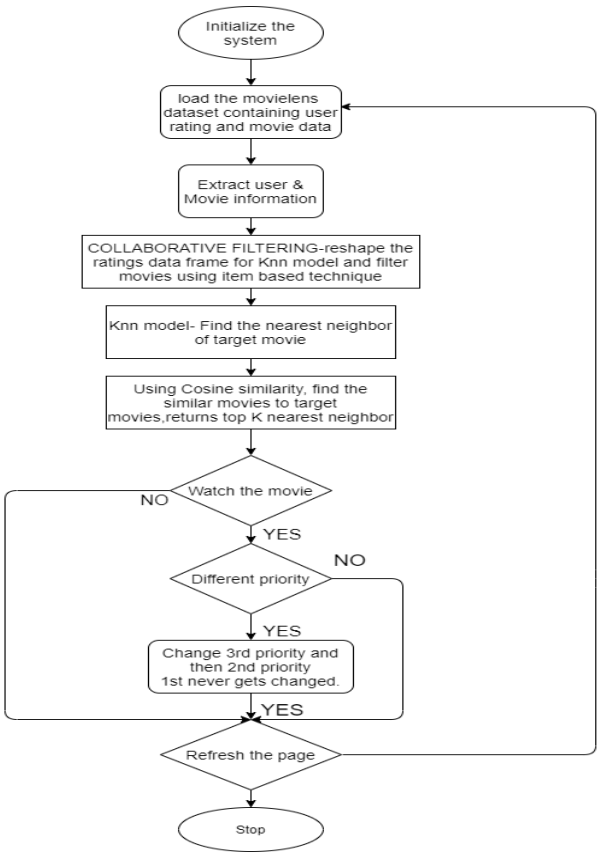

Figure 5: Proposed Collaborative filtering

Figure 5 shows the proposed collaborative filtering method. The objective here is to recommend movies using the item-based technique. First, the extraction of the dataset to gather information about the target movie and the user's rating.

Second, the collaborative filtering begins with the formatting of the rating dataset so that it can be consumed by the KNN model, to remove the huge dataset handling problems. The dataset is reduced according to the popularity removing the noisy error pattern to get the sparse matrix.

Now cosine similarity is used to find the distance between the target movie and other movies, which gives us the top k nearest neighbor. And finally displaying the required recommended list of movies with descending order of distance

In the KNN algorithm, if the value of K=1, then the case is assigned to its nearest neighbor of that class. A case in KNN is classified by the most majority vote of its neighbors, where the case is being allocated to the class most common among its nearest neighbors measured by a distance function.

IV. IMPLEMENTATION

Dataset Description- Movie lens dataset is used containing 28M ratings, over 1M tag application and 60k movies

There is two input database:

1. Movies- containing movie-id, genres, title, user-id in Table 2
2. Rating- containing user-id, rating count, timestamp, movie-id in Table 3.

Filtering of the datasets is done on the basis only to popular ones by filtering the data frame to popular movies only.

Table 2: Movie.csv

S. No	Movie Id	Title
0	1	Toy Story (1995)
1	2	Jumanji (1995)
2	3	Grumpier Old Men (1995)
3	4	Waiting to Exhale (1995)
4	5	Father of the Bride Part II (1995)

Table 3: Rating.csv

S.NO	USERID	MOVIEID	RATING
0	1	1	4.0
1	1	3	4.0
2	1	6	4.0
3	1	47	5.0
4	1	50	5.0

For data analysis, python libraries are used to analyze the movie dataset (as shown in figure 7) and to gain insight into the dataset that helped us in building the module. Every user has rated at least 20 movies. The use of pandas and NumPy and ScikitLearn, scipy libraries were for efficient results on Jupyter Notebook python in this proposed approach. Every movie is rated in a range of 1 to 5(5 being the highest). Figure 6 shows, maximum rating of 3 and 4 for the movie in respect to other scores.

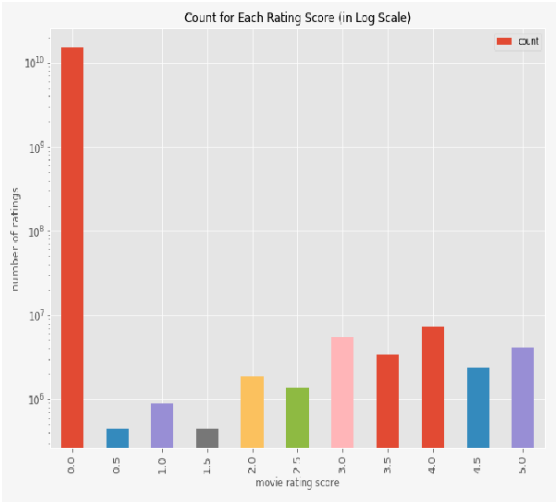

Figure 6: Count for each rating score

```
import pandas as pd
import NumPy as np
from scipy.sparse import csr_matrix
from sklearn.neighbors import NearestNeighbors
```

Figure 7: Library used

As the dataset is huge and there can be where most of the movies are not rated or rated by only one, considering that, a sparse pivot matrix table is developed to transform the data frame into a proper data frame which can be further implemented by KNN and filing 0's in missing information fields.[13]

The KNN analyses the pivot table and uses cosine similarity to find the similarity with the target movie and shows the result.

V. RESULT ANALYSIS

The distance parameter evaluates the similarity or distance between the chosen movie with the others, and the distance is displayed in descending order were the first recommendation.

1: Clear and Present Danger (1994), with distance of 0.3035197854042053:

2: Cliffhanger (1993), with distance of 0.3638320565223694:

3: Fugitive, The (1993), with distance of 0.38245856761932373:

4: Firm, The (1993), with distance of 0.3860310912132263:

5: Outbreak (1995), with distance of 0.3874228596687317:

Figure 8 (a): Recommendation List: Recommendations for Crimson Tide (1995)

For example, the movie Crimson tide (1995), gives this result in figure 8 (a).

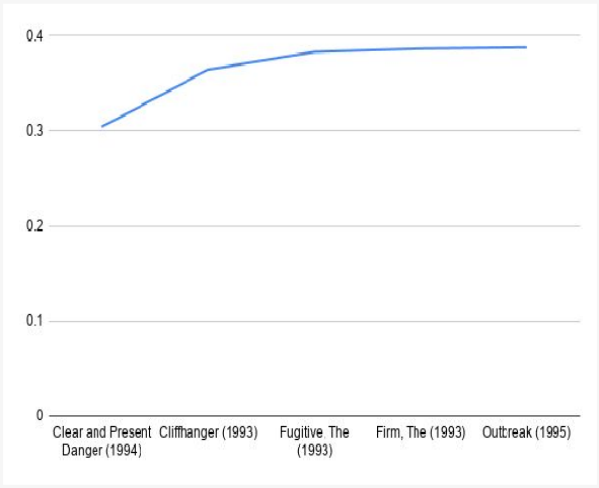

Figure 8 (b): Recommendation list graph

Figure 8 (b), shows the recommendation is based on similarity distance are lesser than the distance, more the similarity with target movie

To evaluate the performance of the proposed system and to provide better results, experiments are conducted by comparing any random existing system with our proposed system based on terms of quality, accuracy, precision, recall, time computation.

The accuracy-numeric value which determines the result of calculation fulfills to the precise or standard value. The metric used are discussed below from Eq. (2) to Eq. (5):

- Precision- is the ratio of recommended items that are relevant to the total number of items on the list [11] [12].

$$Precision = \frac{True\ positive}{True\ positive + False\ Positive} \quad Eq.(2)$$

- Recall is basically describes the relevant prediction from the list of predictions.

$$Recall = \frac{True\ positive}{True\ positive + False\ Negative} \quad Eq.(3)$$

- F-measure (F1) is the scores of the harmonic mean of precision and recall.

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad Eq.(4)$$

- Mean Absolute Error (MAE) is inversely proportional to the performance of the system, also known as Mean Median Regression without considering the directions, it measures the average magnitude of errors in a list of the set of predictions [11].

$$MAE = absolute * \frac{Real\ Value}{Predicted\ Value} \quad Eq.(5)$$

The results in Table 4 show that our model Collaborative filtering works better and shows better results than the regular content-based filtering approach. As the value of the TP rate is stronger than that of the content-based approach, proving the accuracy is improved. The MAE value of our proposed system is 0.248 which is lesser than other recommendation systems as shown in table 5.

Table 4: Performance of recommendation algorithms

Approach	TP rate	Precision	F1
Content-based	0.591	0.501	0.528
Collaborative based(Item-item)	0.761	0.782	0.772

Table 5: Comparison of MAE values

Method	Algorithm	MAE
Content-based	TD-IDF	0.269
Collaborative	MODEL-BASED	0.265
Collaborative	USER-USER	0.258
Collaborative	ITEM-ITEM	0.248

VI. CONCLUSION

In this paper, to avoid the use of content-based filtering, the Item-based CF filtering approach is used for obtaining better results. KNN collaborative recommendation system is proposed using cosine similarity by employing Movielens dataset containing 28M rating for over 60K movies. The existing system are compared and found that the proposed system is more reliable and accurate. It is also found that when the proposed methodology is applied to different larger datasets, both accuracy, and efficiency increase which proves that our system is both accurate and as well as efficient. This item-based filtering is more convenient than user-based. The main aim was to improve the regular recommendation algorithm and to provide better results. The research work was successful as it has been able to fulfill our aim of the project. In the future, more features can be included to datasets (year of release, actor, genre, casting details, etc) to make recommendations more reliable and innovative. The content-based filtering and collaborative filtering can be combined to minimize the errors and improve the performance as a hybrid approach.

REFERENCES

1. C. S. M. Wu, D. Garg, and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 11-15, IEEE, 2018 Nov.
2. R. E. Nakhli, H. Moradi, and M. A. Sadeghi, "Movie Recommender System Based on Percentage of View," In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, pp. 656-660, IEEE.
3. H. W. Chen, Y. L. Wu, M. K. Hor, and C. Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," In *2017 International Conference on Machine Learning and Cybernetics (ICMLC)* , vol. 2, pp. 504-509, Jul, 2017, IEEE.
4. P. Phorasim and L. Yu, "Movies recommendation system using collaborative filtering and k-means," *International Journal of Advanced Computer Research*, vol. *7, no.* 29, p.52, 2017.
5. M. Bahadorpour, B. S. Neysiani, and M. N. Shahraki, "Determining Optimal Number of Neighbors in Item-based kNN Collaborative Filtering Algorithm for Learning Preferences of New Users," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. *9, no.* 3, pp.163-167, 2017.
6. A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," In *2016 4th International Conference on Cyber and IT Service Management*, pp. 1-6, IEEE, Apr, 2016.
7. D. Pathak, S. Matharia, and C. N. S. Murthy, "ORBIT: Hybrid movie recommendation engine," In *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, pp. 19-24, IEEE, Mar, 2013.
8. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295, Apr, 2011.
9. R. Van Meteren, and M. Van Someren, "Using content-based filtering for recommendation," In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, vol. 30, pp. 47-56, May 2000.
10. L. E. Molina and S. Bhulai, Recommendation System for Netflix, 2018.
11. P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems." In: Proceedings of the fifth ACM conference on Recommender Systems (RecSys'11), ACM, New York, NY, USA, pp. 157-164, 2011.
12. M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation measures for models assessment over imbalanced data sets," *J Inf Eng Appl,* vol. 3, no. 10, 2013.
13. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295).
14. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (1994, October) (pp. 175-186).
15. J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, " GroupLens: applying collaborative filtering to Usenet news," *Communications of the ACM*, vol. *40, no.* 3, pp. 77-87.
16. W. Liang, G. Lu, X. Ji, J. Li, and D. Yuan, "Difference factor' KNN collaborative filtering recommendation algorithm," In *International Conference on Advanced Data Mining and Applications* (2014, December). Springer, Cham, pp. 175-184.
17. J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98),* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 43–52.