

Deep Recommender Systems with Python



Panagiotis Symeonidis
Associate Professor
University of the Aegean

Deep Recommender Systems with Python

Data Mining, Information Retrieval, Machine Learning and Artificial Intelligence Algorithms

Panagiotis Symeonidis

Associate Professor – University of the Aegean

psymeon@aegean.gr

Dedicated to my son **Dimitris!**

(This pages is left intentionally blank)

Contents

- Foreword** 13

- 1 Introduction to Recommender Systems** 15

 - 1.1 Objectives of a Recommendation System** 15
 - 1.1.1 Objectives concerning the Users 15
 - 1.1.2 Objectives concerning the Service Providers 16

 - 1.2 Types of Recommender Systems** 16
 - 1.2.1 Collaborative Filtering 16
 - 1.2.2 Demographic Filtering 17
 - 1.2.3 Content-based Filtering 17
 - 1.2.4 Knowledge-based Systems 18
 - 1.2.5 Hybrid Systems 19

 - 1.3 Types of Recommendation Algorithms** 19
 - 1.3.1 Memory-based algorithms 20
 - 1.3.2 Model-based algorithms 21
 - 1.3.3 Data Mining Algorithms 23
 - 1.3.4 Information Retrieval Algorithms 24

1.3.5	Machine Learning Algorithms	27
1.4	Chapter Questions	28
	Bibliography	29
2	Collaborative Filtering Algorithms	31
2.1	Introduction	31
2.2	Related Work	33
2.3	Factors Affecting the Collaborative Filtering Process	33
2.3.1	First Stage Factors	35
2.3.2	Second Stage Factors	37
2.3.3	Third Stage Factors	40
2.4	Comparative Performance Study of CF Algorithms	41
2.5	Conclusions	41
2.6	Chapter Questions	42
2.7	Python Programming Exercises	43
2.7.1	Exercise 1: Constructing a User-User Similarity Matrix in Python	43
	Bibliography	49
3	Content-based Filtering Algorithms	51
3.1	Introduction	51
3.2	Related Work	52
3.3	The Vector Space Model (Vector Space Model)	53
3.4	Recommendations based on Content Data	53
3.4.1	User profile construction step	54
3.4.2	Feature Weighting Step	55
3.4.3	User's Neighborhood Formation Step	57
3.4.4	Item Recommendation List Creation step	57
3.5	Decision Tree Classifier	58
3.5.1	Creating Decision Trees based on Collaborative Filtering	60

3.6	Feature Selection	61
3.6.1	Gini index	61
3.6.2	Entropy	61
3.6.3	X^2 -statistic	64
3.7	Naive Bayes classifier	65
3.8	Chapter Questions	67
3.9	Python Programming Exercises	69
3.9.1	Recommend new friends based on a content-based filtering algorithm	69
	Bibliography	73
4	Context-aware Recommender Systems	75
4.1	Time-aware Recommendation Systems	75
4.1.1	Introduction	75
4.1.2	Session-based Recommendation Systems	76
4.1.3	Defining an Evaluation Protocol for Session-based Recommender Systems	81
4.1.4	Comparison of Evaluation Protocols: Time-dependent vs. Time-independent	82
4.1.5	The Challenges of Time-dependent Evaluation Protocol.	83
4.2	Location-aware Recommendation Systems	84
4.2.1	Introduction	84
4.3	Recommendation systems for LBSNs	87
4.4	Types of Recommendations in LBSNs	88
4.4.1	Friend Recommendation	89
4.4.2	Point of Interest (POI) Recommendation	89
4.4.3	Route/Itinerary Recommendation	89
4.4.4	Activity Recommendation	90
4.4.5	Event Recommendation	90
4.5	Hybrid and Ensemble Methods	91
4.5.1	Hybrid Methods	91

4.5.2	Ensemble Methods	91
4.6	Types of Hybrid Systems	91
4.6.1	Hybrid Monolithic Systems	92
4.6.2	Hybrid Pipelined Systems	92
4.6.3	Hybrid Parallelized Systems	93
4.7	Chapter Questions	96
4.8	Python Programming Exercises	97
4.8.1	Exercise 1: Recommend new friends based on friendship links of a graph.	97
4.8.2	Exercise 2: Recommend new friends based on a hybrid recommendation algorithm	99
	Bibliography	101
5	Matrix Factorization Algorithms	103
5.1	Introduction	103
5.2	Eigenvalue Decomposition	104
5.3	Singular Value Decomposition	107
5.3.1	Applying the SVD Factorization	109
5.3.2	Preserving the largest c -Singular Values	110
5.3.3	Inserting a Test User in the c -dimensional Space	111
5.3.4	Generating the Neighborhood of Users/Items	112
5.3.5	Generating the Recommendation List	112
5.4	From SVD to UV-decomposition	112
5.4.1	Objective Function Formulation	115
5.4.2	Avoiding Overfitting with Regularization	116
5.4.3	The UV-decomposition algorithm	118
5.4.4	Step-by-Step Computation of UV-Decomposition	119
5.5	Tensor Decomposition (Tensor decomposition)	123
5.6	Tucker Decomposition and HOSVD	125
5.7	AlSHOSVD Algorithm	127

5.8	Bayesian Pairwise Ranking Algorithm	128
5.9	List-wise Ranking Algorithm	130
5.10	Chapter Questions	131
5.11	Python Programming Exercises	132
5.11.1	Apply UV-decomposition to a matrix using Python.	132
	Bibliography	136
6	Deep Neural Networks and Genetic Algorithms	139
6.1	The structure of a Perceptron	139
6.1.1	Single-layer Perceptron	139
6.1.2	Multi-layer Perceptron	146
6.2	Convolutional Neural Networks	148
6.3	Recurrent Neural Networks	150
6.4	Genetic Algorithms	151
6.4.1	The Structural Components of Genetic Algorithms	151
6.4.2	Genetic Operations	152
6.4.3	Related Work	155
6.4.4	The Architecture of a Genetic Algorithm for Recommendation Systems	155
6.4.5	The Genetic Algorithm in pseudocode form	157
6.4.6	Step-by-Step Execution of the Genetic Algorithm	158
6.5	Chapter Questions	161
6.6	Python Programming Exercises	162
6.6.1	Factorize the user-item rating matrix by using a neural network with Python.	162
	Bibliography	172
7	Deep Reinforcement Learning for Recommender Systems	175
7.1	Markov Chain Model	175
7.1.1	Markov Decision Processes	176

7.2	Reinforcement Learning Algorithms	178
7.3	Q-learning Algorithm	180
7.3.1	The Tabular Q-learning Algorithm in Pseudocode	182
7.3.2	Step-by-Step Execution of the Tabular Q-learning Algorithm	184
7.4	Deep Reinforcement Learning	186
7.4.1	The Deep Q-Network (DQN) with Experience Replay Algorithm in Pseudocode . . .	187
7.4.2	Advantage Actor Critic (A2C) Algorithm	189
7.5	Chapter Questions	191
7.6	Python Programming Exercises	192
7.6.1	Exercise 1: Implement the Tabular Q-learning algorithm for the rating prediction task. 192	
7.6.2	Exercise 2: Implement the DQN with experience replay algorithm for the rating prediction task.	198
	Bibliography	209
8	Deep Graph Neural Networks for Recommender Systems	211
8.1	Introduction to Heterogeneous Graphs: Fundamentals	211
8.1.1	Heterogeneous Information Network Definition	211
8.1.2	Network Schema Definition	212
8.2	Related Bibliography	212
8.3	Local-based Similarity Measures	214
8.3.1	Shortest Path	214
8.3.2	Common Neighbors	214
8.3.3	Jaccard similarity index	214
8.3.4	Salton similarity index	214
8.3.5	Adamic & Adar similarity index	215
8.3.6	Preferential Attachment	215
8.3.7	FriendLink	215

8.4	Global-based Similarity Algorithms	216
8.4.1	Katz Status Index	216
8.4.2	SimRank for Unipartite Graphs	219
8.4.3	SimRank Algorithm for Bipartite Graphs	222
8.4.4	Random Walk with Restart algorithm	224
8.4.5	PathSim Algorithm	227
8.4.6	Explanation of recommendations based on meta-paths	229
8.5	Knowledge Graphs	231
8.6	Graph Convolutional Networks	233
8.6.1	Graph Attention Mechanism	237
8.6.2	Node representation in GCNs by using Linear Algebra Matrices	238
8.6.3	Loss Function Definition for Predicting Ratings in GCNs	238
8.7	Graph Embeddings	239
8.8	Chapter Questions	240
8.9	Exercises in Programming	241
8.9.1	Heterogeneous Information Network for an Online Newspaper	241
	Bibliography	248
9	Evaluation Metrics	251
9.1	Introduction	251
9.2	MAE and RMSE metrics	253
9.3	Confusion Matrix	255
9.4	Precision, Recall, and F1 metric	256
9.5	Precision-Recall curve and the MAP metric	257
9.6	ROC curve, and AUC metric	260
9.7	Normalized Discounted Cumulative Gain	262
9.8	Beyond Accuracy Metrics	264
9.8.1	Novel Recommendations	264

9.8.2	Diversified Item Recommendations	268
9.8.3	Explainable Recommendations	268
9.8.4	Item Style of Explanation	271
9.9	Chapter Questions	272
9.10	Programming Exercises	274
9.10.1	Create a framework for evaluating a recommendation algorithm by using Python	274
	Bibliography	286
10	New Trends	289
10.1	Group Recommender Systems	289
10.1.1	Group Members' Ratings Aggregation	290
10.1.2	Ranking Lists' Aggregation	292
10.2	Ethics in Recommender Systems	296
10.2.1	Fairness	296
10.2.2	Accountability	300
10.2.3	Censorship	300
10.3	Privacy and Recommendation Systems	301
10.3.1	Systems Architecture for Privacy	302
10.3.2	Algorithmic Techniques for Privacy Protection	303
10.3.3	Legal Framework for the Protection of Privacy	314
10.4	Chapter Questions	315
	Bibliography	315
	Total Bibliography of all chapters	317
	Index	319

Foreword

This book covers a relatively new scientific field, which has not received the required attention in the Computer Science, Computer Engineering and Informatics Departments/Schools. In this work, therefore, we present in detail the most important algorithms of the last 20 years from the main scientific fields of Information Retrieval, Machine Learning, Data Mining and Artificial Intelligence, which have influenced and shaped Recommender Systems. The timing of this publication can be described as mature, since the author has been doing research work for almost 20 years (since 2005) in the scientific field of *Recommender Systems*.

Furthermore, regarding the theoretical part of the course *Deep Recommender Systems*, the presentation of the algorithms is done in an educational and pedagogical way, in order to enable students to better comprehend the different concepts and methods. In particular, the book contains more than 40 solved examples, so that the mathematical formulas and equations of the theory are applied to the data of the examples in order to be fully understood by the student. Also, the same example (with data from a few users who have rated a few items) runs through several different chapters, so that the student has a reference and comparison point between the different algorithms applied to it and can follow their step-by-step execution. For this reason, the book can be taught at undergraduate level in addition to graduate level. For the outline of an undergraduate course, chapters 1 to 4 are suggested, as well as chapters 8 to 10, which introduce the student to the field of *Recommender Systems*. Similarly, for the outline of a postgraduate course, chapters 1 and 2 are suggested, as well as chapters 5 to 10, which describe more advanced topics of *Deep Recommender Systems*. It is emphasized that the slides for each chapter can be downloaded from the author's official website, <http://www.panagiotissymeonidis.com/enbook>.

Apart from the theoretical part, the book can also support the laboratory part of the aforementioned course *Deep Recommender Systems*. Therefore, the most basic algorithms for implement-

ing a recommender system are described with *Python* language (using *Notebook Jupyter*) in a total of 8 solved programming exercises at the end of each chapter. Besides, the code of the algorithms is described step-by-step in the respective sections, so that the reader can verify all intermediate results of the code. Also, the code can be downloaded from the author's official website, <http://www.panagiotissymeonidis.com/enbook>. Finally, the algorithm code can be executed *directly (online)* at <https://colab.research.google.com> without requiring any installation on the student's computer.

The book is also suitable for researchers and professionals in the field of *Computer Science*, since the advanced topics that form the field of *Deep Recommender Systems* are covered. It additionally provides researchers and developers with a comprehensive overview of the fundamental concepts of *Deep Recommender Systems*, and presents all new methods and techniques through realistic and modern examples.

At this point, I would especially like to thank my colleague and friend Christos Andras, who provided useful comments for the book's further improvement. Furthermore, I would like to thank all my undergraduate and graduate students who helped with their comments in proofreading the book. Finally, I thank my colleagues Yannis Manolopoulos, and Efstathios Stamatatos, for their helpful comments and suggestions.