# Recommendation Systems summary

A. Gidiotis, K. Stamatiadis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 1.0

Artificial Intelligence &
Information Analysis Lab

# Recommendation Systems

- **Content Based Filtering**
- Collaborative Filtering
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- Matrix Factorization
- Hybridization techniques
- Deep Learning in Recommender Systems
- Evaluation of Recommender Systems
- Netflix Challenge

# Motivation

- One of the major machine learning applications in industry.
- Immediate impact on companies.
- Responsible for a substantial fraction of revenue.
- Netflix, Amazon, Spotify etc.

Artificial Intelligence & Information Analysis Lab

# Problem Formulation

- Supposing running an online movies company. The goal is to recommend to the users new movies they will probably enjoy.
- For each user we know some movies he has already seen and enjoyed.
- Each user rates movies giving them 1-5 stars.
- What we want is to make predictions of the rating of a user to a new movie based on his movie and rating history.
- The movies recommended are the ones with the highest rating predictions.

# Content Based Systems

- Recommendations made by using a user’s item and profile features.
- Construction of groups with similar items based on their features
- Creation of user profiles based on past interactions and on user’s interests.

# Content Based Recommendations

- We assume that for each movie we know some features {$\boldsymbol{\theta}^1$, $\boldsymbol{\theta}^2$, …, $\boldsymbol{\theta}^j$}.
  - e.g. the movie involves romance, action etc.
- We try to estimate a set of user features {$\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^i$}.
  - e.g. the user likes action but dislikes romance.
- The expected rating of user i for movie j $r^{i,j}$ is given by the dot product:

$$r^{i,j} = (\theta^j)^T \cdot x^i.$$

# Learning Users

- In order to estimate the user features $\mathbf{x}^i$ we can try to minimize the squared error of the predictions $\mathbf{r}^{i,j}$ given the targets $\mathbf{y}^{i,j}$:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j:r(i,j)=1}((\theta^j)^T \cdot x^i - y^{i,j})^2 + \frac{\lambda}{2}\sum_{i=1}^{\mathrm{N}}\sum_{j=1}^{K}(\theta^{i,j})^2 .$$

Artificial Intelligence & Information Analysis Lab

# Recommendation Systems

- Content Based Filtering
- **Collaborative Filtering**
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- Matrix Factorization
- Hybridization techniques
- Deep Learning in Recommender Systems
- Evaluation of Recommender Systems
- Netflix Challenge

# Collaborative Filtering

- What if we don't know the movie content?
- Feature engineering is usually hard.
- We can learn movie features and user features simultaneously.

Artificial Intelligence & Information Analysis Lab

# Collaborative Filtering

- One of the most frequently used approaches.
- In general it is more effective than collaborative filtering
  - Memory based systems
  - Model based systems

- Techniques used in collaborative filtering systems:
  - Fully-connected neural networks.
  - Item2vec.

# Memory-Based Systems

- Two approaches:
- By identifying clusters of users and utilizing the interactions of one specific user to predict the interactions of other similar users.
- By identifying clusters of items that have been rated by user A and utilizing them to predict the interaction of user A with a different but similar item B.

# Model-Based Systems

- Based on machine learning and data mining techniques.
- Train models to be able to make predictions.

# Collaborative Filtering

- The new cost function becomes:

$$\mathrm{J}(\theta, x) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j:r(i,j)=1}\left((\theta^{j})^{T}\cdot x^{i} - y^{i,j}\right)^{2} + \frac{\lambda}{2}\sum_{i=1}^{\mathrm{N}}\sum_{j=1}^{K}(\theta^{i,j})^{2}.$$

# Alternating Least Squares

The ALS algorithm is the most commonly used algorithm that solves the collaborative filtering problem.

- Two steps:
  1) Freeze the movie features and learn the user features for some iterations.
  2) Freeze the new user features and learn the movie features.

- These steps are repeated until convergence.

# Implicit Datasets

- In many cases our data may not be user ratings.
- We may have instances of the form:
  - user **i** interacted with movie **j**, **n** times.
- Key differences:
  - We do not have missing values.
  - We cannot assume that if a user, item pair has a value of zero then the user did not like the item.
  - We do not have explicit likes/dislikes but instead confidence.

# Distributed Implementation

- The ALS algorithm can easily be parallelized and implemented efficiently using Hadoop map-reduce.

- Apache Spark ML library implements ALS algorithm among others.

# Recommendation Systems

- Content Based Filtering
- Collaborative Filtering
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- **Matrix Factorization**
- Hybridization techniques
- Deep Learning in Recommender Systems
- Evaluation of Recommender Systems
- Netflix Challenge

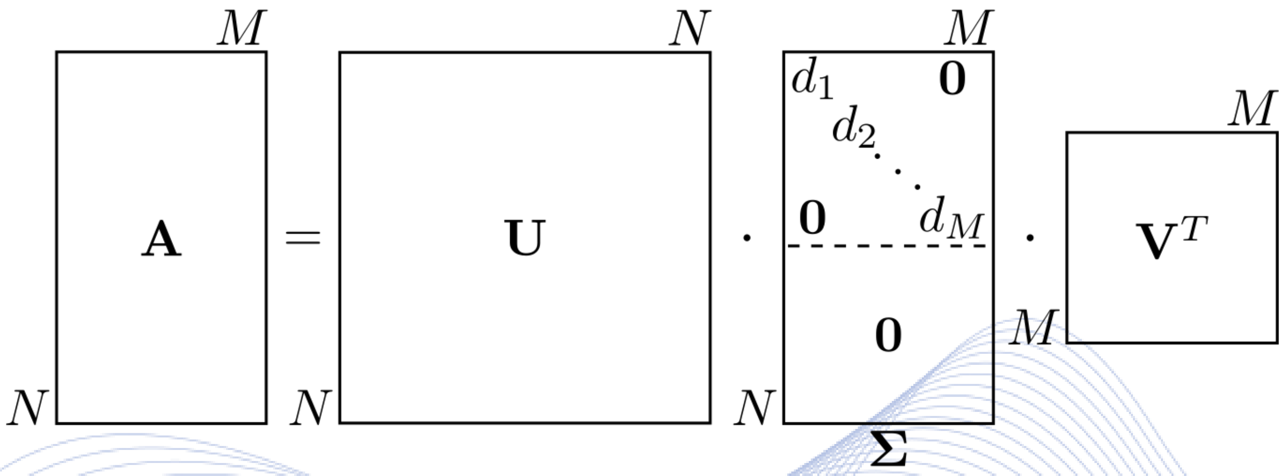
# Matrix Factorization

Any $n \times m$ user-item rating matrix $\mathbf{R}$ can be decomposed into:

$$\mathbf{R} = \mathbf{U\Sigma V}^T,$$

- $\mathbf{U}$ ($n \times n$ orthogonal) unitary matrix,
- $\mathbf{\Sigma}$ ($n \times m$ diagonal) matrix and
- $\mathbf{V}^T$ ($m \times m$ orthogonal) unitary matrix.
- ***Singular values*** of $\mathbf{R}$ are the $r = \min(n, m)$ $\sigma_1, \sigma_2, \dots, \sigma_r$ diagonal elements of $\mathbf{\Sigma}$.

# Matrix Factorization




Singular Value Matrix Decomposition of a user-item rating matrix.

# Matrix Factorization

- Two lower dimension rectangular matrices are produced after the decomposition of the user/item interaction matrix.

- In order to make improvements to the results of the prediction, we assign different weights to the latent factors. The weights are affected by the amount of popularity among the items, as well as from users' amount of activity.

# Funk MF

- The user-item matrix is being factorized by two matrices of lower dimension. Each row in the first matrix represents a user and each column in the second matrix represents an item.

- We call *latent factors* rows or columns that are associated with a specific item or a user.

- SVD-like machine learning model.

# SVD++

- Funk MF uses only explicit numerical ratings.
- New recommender systems must take advantage both explicit and implicit interactions.
- In comparison with Funk MF, SVD++ takes into consideration user and item bias

# Asymmetric SVD

- Asymmetric SVD is a model-based algorithm. Its basic concept is to combine SVD++'s advantages.

- Q replaces the user latent factor matrix H. The user's preferences are being learned as a function of their ratings:

$$\tilde{r}_{ui} = \mu + b_i + b_u + \sum_{f=0}^{n\,factors} \sum_{j=0}^{n\,items} r_{uj} Q_{j,f} W_{f,i}\,.$$

Artificial Intelligence & Information Analysis Lab

# Recommendation Systems

- Content Based Filtering
- Collaborative Filtering
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- Matrix Factorization
- **Hybridization techniques**
- Deep Learning in Recommender Systems
- Evaluation of Recommender Systems
- Netflix Challenge

# Hybrid Systems

- Implemented in several ways:
  - By creating separate content based and collaborative based predictions, and finally combining them.
  - By the addition of collaborative based features to a content based approach and vice versa
  - By creating one model with both approaches blended in

- Cold start and the sparsity problem can also be dealt with hybrid systems.

# Hybrid Systems & Netflix

- Netflix uses hybrid recommender system
    - Use of collaborative filtering. It makes comparisons between the most frequent searches and views of similar users.
    - The movies recommended have common characteristics with movies that a user likes (content-based filtering).

# Recommendation Systems

- Content Based Filtering
- Collaborative Filtering
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- Matrix Factorization
- Hybridization techniques
- **Deep Learning in Recommender Systems**
- Evaluation of Recommender Systems
- Netflix Challenge

# Motivation

- Collaborative filtering with sparse data is a difficult non-convex problem.
- SVD might not be able to solve this by minimizing the squared error if the data are too sparse.
- SVD and ALS are linear models.
- We can do better by using deep non-linear models.

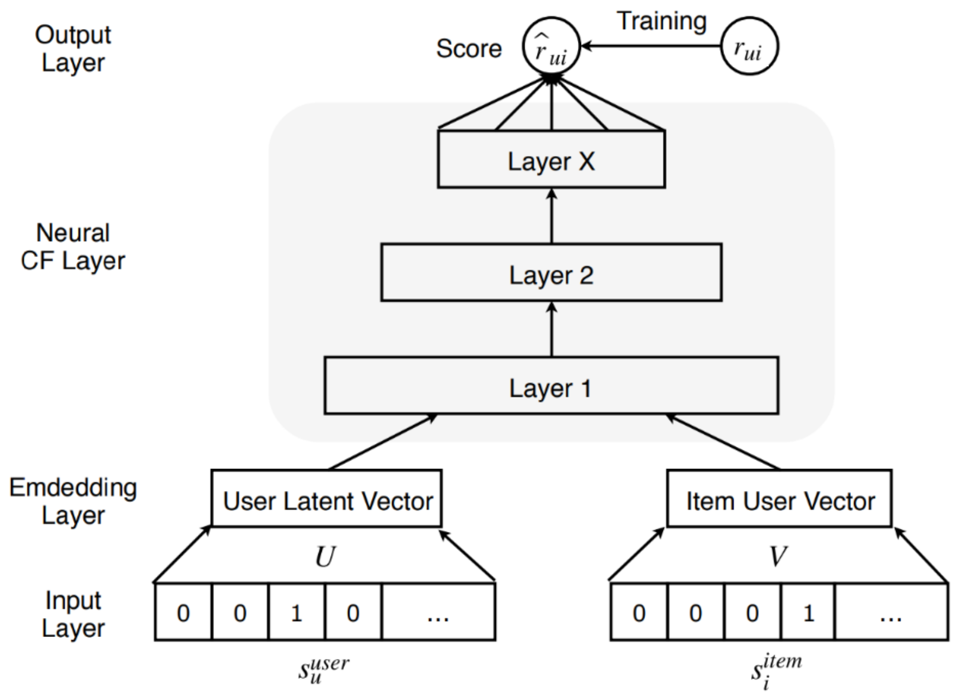
# Multilayer Perceptron based Recommendation

- MLP is the basis of numerous advanced approaches.

- Can be similar to any function to any desired accuracy level.

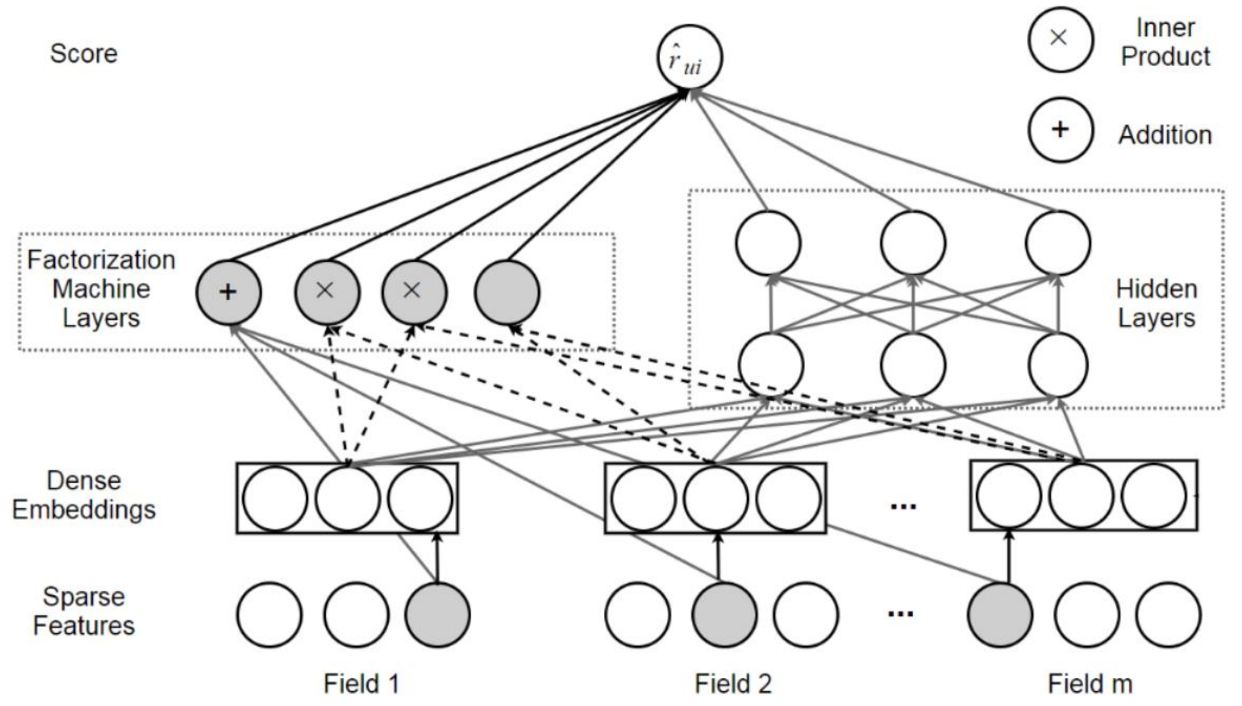
# Neural Collaborative Filtering

- Illustration of Neural Collaborative Filtering.




Artificial Intelligence & Information Analysis Lab

# Deep Factorization Machine

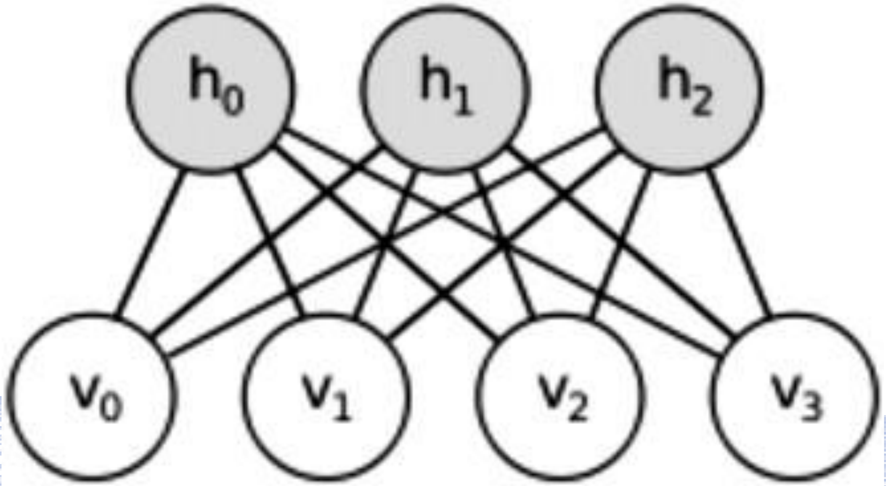



Deep Factorization Machine.

Artificial Intelligence & Information Analysis Lab

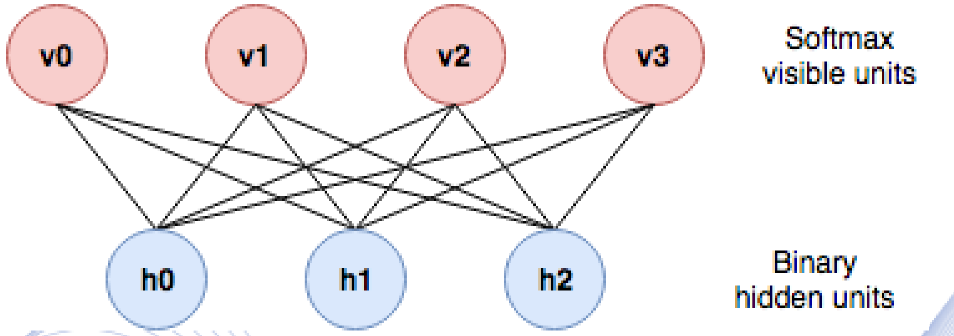
# Restricted Boltzman Machines

- RBMs are undirected graphs.
- They have a set of visible units and a set of hidden units.
- Can learn to model distribution in an unsupervised way.
- They can be good feature extractors.
  - Learn to model distributions by observation.



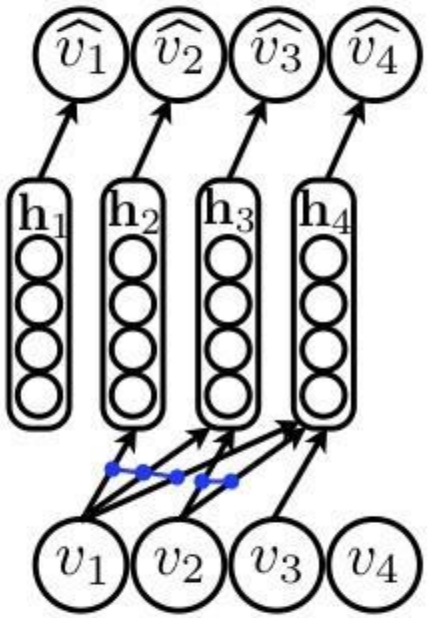
# RBMs for Collaborative Filtering

- In the simple case we assume that M users have rated all N movies.
- We create a single RBM that models all.
- The RBM has M visible softmax units (ratings 1-5).
- The RBM has F hidden units.
- The hidden units learn dependencies between ratings of different movies.
- Each user is a single example for the RBM.



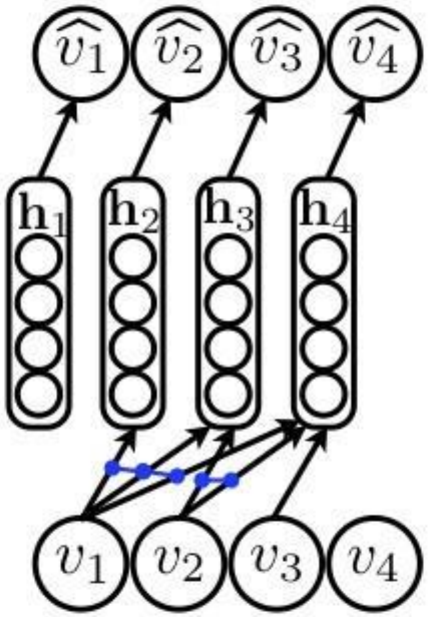
# Neural Autoregressive Density Estimators

- Neural Autoregressive Density Estimators (NADE) are inspired by RBMs.
- Tractable distribution estimators.
- We can exactly and efficiently compute probabilities as well as sample new observations.
- Can be efficiently trained with backpropagation.



# NADE for Collaborative Filtering

- We use a CF-NADE for each user.
- Shared parameters between users that rated the same movie.
- Same number of hidden units.
- Only D << N visible units.
- One training case per CF-NADE/user.



# Recommendation Systems

- Content Based Filtering
- Collaborative Filtering
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- Matrix Factorization
- Hybridization techniques
- Deep Learning in Recommender Systems
- **Evaluation of Recommender Systems**
- Netflix Challenge

# Evaluation of Recommender Systems

- The choice for the evaluation way depends on the type of the recommendation that we want to provide.

- The most common types of recommender system evaluations: online and offline approaches.

# Online Methods

- Based on the recommendations that have been made, user reactions are being measured.

- Certain difficulty for the method to be implemented as the system is required to be already in production.

- Experiments that may fail will definitely affect income as well as the users’ experience.

# Offline Methods

- Ideal for experimental/early stages.

- Two main datasets for training and validation.

- Amounts of data are used for the construction of the system as well as for the evaluation process.

# Recommendation Systems

- Content Based Filtering
- Collaborative Filtering
  - KNN algorithm
  - ALS algorithm
  - Learning from Implicit Datasets
- Matrix Factorization
- Hybridization techniques
- Deep Learning in Recommender Systems
- Evaluation of Recommender Systems
- **Netflix Challenge**

# Netflix Challenge

- A *training* data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies was provided by Netflix
- The goal was to give a 10% boost to the accuracy of the recommendation engine.
- Over 2,817,131 triplets of the form <**user**, **movie**, **date of grade**> were contained into the qualifying data set. Grades were known only to the jury.
- Every participant team knows the score only for the half of the data. They should produce an algorithm that must predict grades on the entire set.
- The other half is used as a test set. Based on the performance, the jury will determine the winners of the challenge

# Bibliography

[PIT2016] I. Pitas (editor), “Graph analysis in social media”, CRC Press, 2016.
[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

# Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr