INTRODUCTION TO DATA SCIENCE

Rutgers University

14.2 Recommender Systems II

https://rutgers.instructure.com/courses/30716

Slide credits> CMU Data Sconce

14.2 Recommender Systems

- Content-based systems
- Matrix factorization methods
- Neural network-based systems

Content based systems

Major goal in recommendation system is to provide recommendations for items not rated by a user.

For example, if user i did not rate item j, then find a reasonable estimate using existing ratings in the database.

	Bill	**Sandy**	**Alex**	**Andy**
La La Land	2	4	1	
Karate Kid	5		5	1
Silver Lining		5		2
Mission Impossible	4		5	
Dogs in heaven		1		4

Goal. Use the features of each movie to build a linear regression model for each user

Linear regression model

	Bill	Sandy	Alex	Andy
La La Land	2	4	1	
Karate Kid	5		5	1
Silver Lining		5		2
Mission Impossible	4		5	
Dogs in heaven		1		4

Features

X_1 Romantic	X_2 Drama	X_3 Action
0.7	0.2	0
0	0.2	0.8
0.8	0.2	0
0.1	0	0.9
0.2	0.8	0

Goal. Use the features of each movie to build a linear regression model for each user

Notation

Let $x^{(i)}$ be the input vector for movie i - 3 features + bias

Write down the possible input vectors.

X_1 Romantic	X_2 Drama	X_3 Action
0.7	0.2	0
0	0.2	0.8
0.8	0.2	0
0.1	0	0.9
0.2	0.8	0

$x^{(1)}$

$x^{(2)}$

Let $\Theta^{(j)}$ be the parameters of the linear regression model for user j

Goal. Learn the parameters $\Theta^{(j)}$ for each user j

Predict. Rating of movie i and user j as : $(\Theta^{(j)})^T x^{(i)}$

Optimization Objective

To learn parameter $\Theta^{(j)}$ minimize the following loss function.

To Learn all $\Theta^{(j)}$ for j = 1,…m (users) minimize the following function

Finding the $\Theta^{(j)}$ using gradient descent

Update gradient descent as before in linear regression

Once $\Theta^{(j)}$ are learned, then use them to predict missing entries in the ratings table.

Quiz

Suppose we consider a movie rating table where each movie is defined by 4 features.

1. How large is the feature vector?

2. For each user j define parameters of the linear regression model as $\Theta^{(j)}$. Write down the regression model in matrix form.

3. Suppose parameters for Bill is learned as [0 0.5 0.2 -0.3 -0.1], what is the rating estimate for a new movie with feature vector [0.9 0.1 0 0] ?

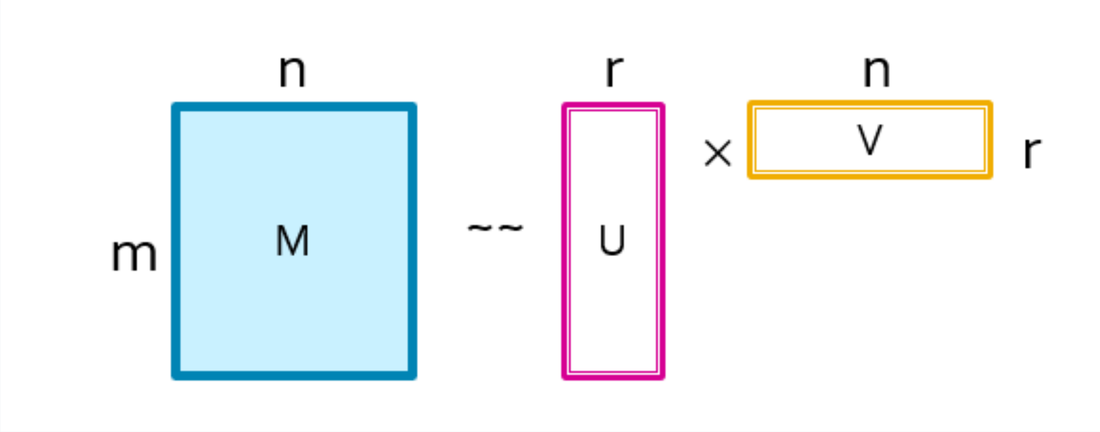
14.2 Recommender Systems

- Content-based systems
- Matrix factorization methods
- Neural network-based systems

Basics of Matrix Factorization

The ratings is represented by a m by n matrix M, where m is the number of users and n is the number of items.

It is possible to factor this matrix into a product of two matrices such that their share a small common dimension.



The small matrices are called low-rank matrices

Data as Relations

The data can exhibit many-many relationships and are represented by a matrix

Examples.

people and movies. where matrix entries are the movie ratings given by people

Student and courses. Where matrix entries are the grades earned by students

Often the relationship can be explained closely by a “latent” factor

Examples

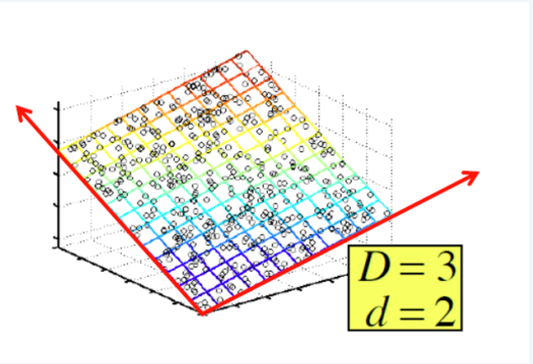
genre of movies – Joe liked La La Land because Joe generally like romantic movies

good at CS – Sandra likes OS because she generally like CS classes

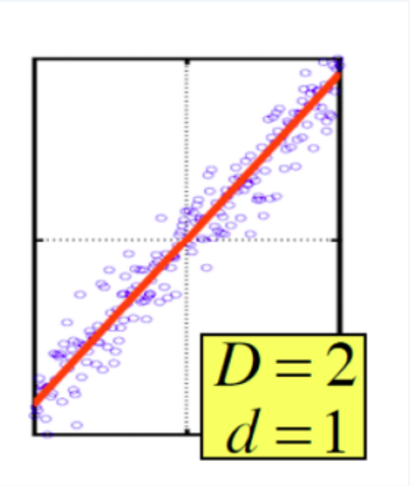
The Objective

Major idea in all this is the dimension reduction

From high dimension



low dimension



The first dimension is along the direction that exhibit the largest variance.

The second dimension is orthogonal to first shows the next best variance

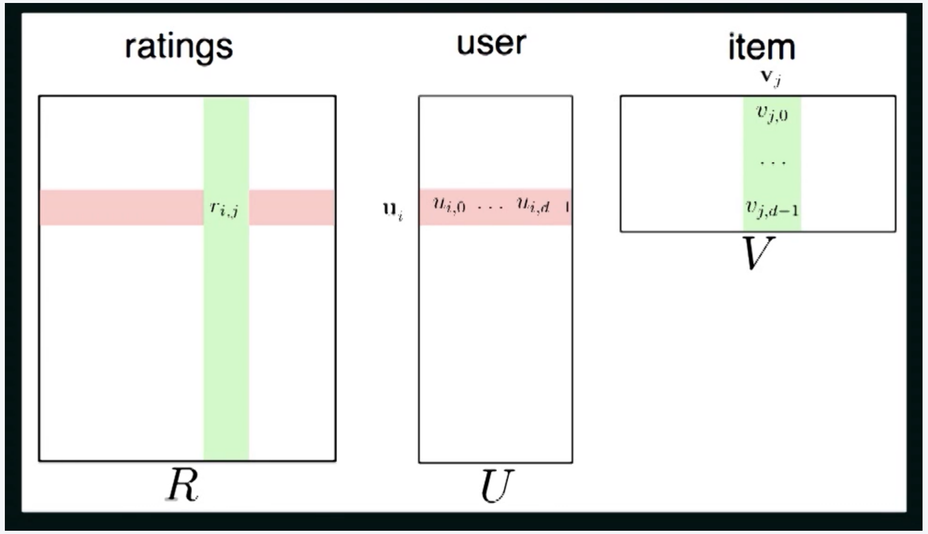
We continue this until we have enough dimensions to capture the features of matrix

Using matrix factorization for rating estimation

Factorize user-item matrix X into a low-rank factorization

$$X \approx UV, \qquad U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{k \times n}$$

$$U = \begin{bmatrix} - & u_1^T & - \\ & \vdots & \\ - & u_m^T & - \end{bmatrix}, \qquad V = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix}$$



Note that U is shown as a row matrix and V is shown as a column matrix

Loss Function

$$L(U,V) = \sum_{(i,j) \in D} ||r_{i,j} - \mathbf{u}_i^T \cdot \mathbf{v}_j||_2^2 + \lambda(||U||_2^2 + ||V||_2^2)$$

Back to canonical ML model

Hypothesis function

user specific

item specific

$$\hat{X}_{ij} \equiv h_\theta(i,j) = u_i^T v_j$$

u, v are lower rank matrices

Loss Function

$$\ell(h_\theta(i,j), X_{ij}) = (h_\theta(i,j) - X_{ij})^2$$

think of parameters Θ as u's and v's

Optimization

$$\underset{u \quad v}{\text{minimize}} \sum_{i,j \in S} (u_i^T v_j - X_{ij})^2$$

where

$$S = \{(i,j) : X_{ij} \neq 0\}$$

Learn the users and items at the same time.

Optimization objective

Optimization objective (single example)

$$\underset{u_i}{\text{minimize}} \sum_{j:(i,j)\in S} (v_j^T u_i - X_{ij})^2$$

Analytical Solution (without proof)

$$u_i = \left(\sum_{j:(i,j)\in S} v_j v_j^T \right)^{-1} \left(\sum_{j:(i,j)\in S} v_j X_{ij} \right)$$

Alternating minimization algorithm: Repeatedly solve for all u_i for each user, v_j for each item (may not give global optimum)

PCA and matrix factorization

PCA is also obtained using a factorization of matrix X

- SVD on X gives $X = U\ D\ V^T$
- $C = = X^T X / n = (V\ D\ U^T)(U\ D\ V^T) / n = V\ (D^2/n)\ V^T$
- Principal components are given by $X\ V = (U\ D\ V^T)\ V = UD$

One major difference between PCA and Collaborative Filtering

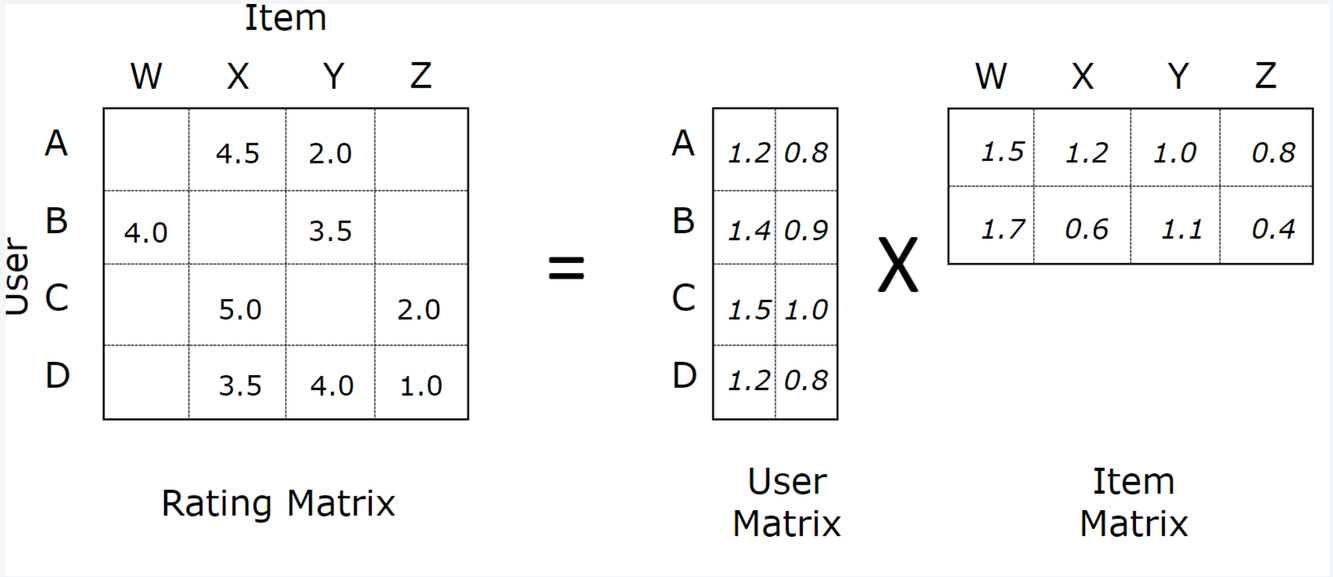
In PCA all entries are observed.

In some ways the PCA and Collaborative filtering are the “same” concept

PCA reduces things that do not matter much, and collaborative filtering does the same thing

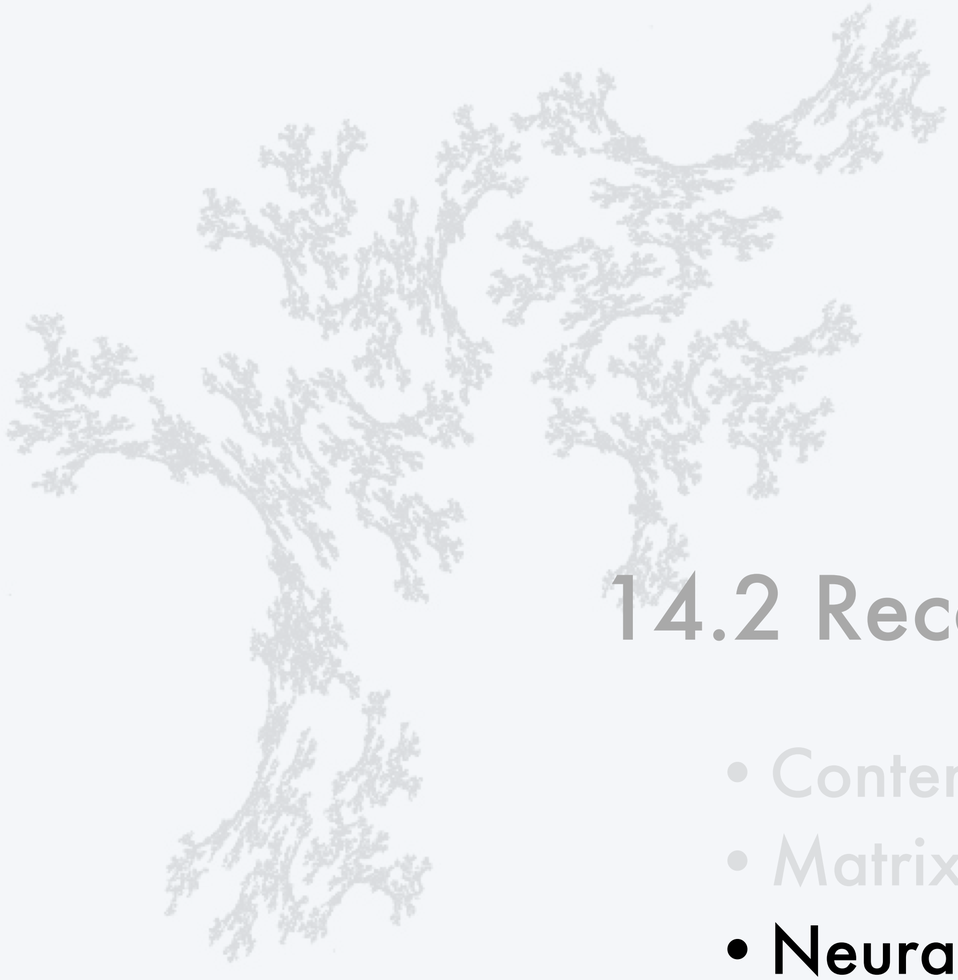
Quiz

Consider the following factorization of a user-item matrix.



Estimate the missing entry (C,W)

Image credit: https://towardsdatascience.com/



14.2 Recommender Systems

- Content-based systems
- Matrix factorization methods
- Neural network-based systems

From categorical to numerical

NN based systems are numerical

But data in recommender system may be categorical due to Large collection of discrete symbols

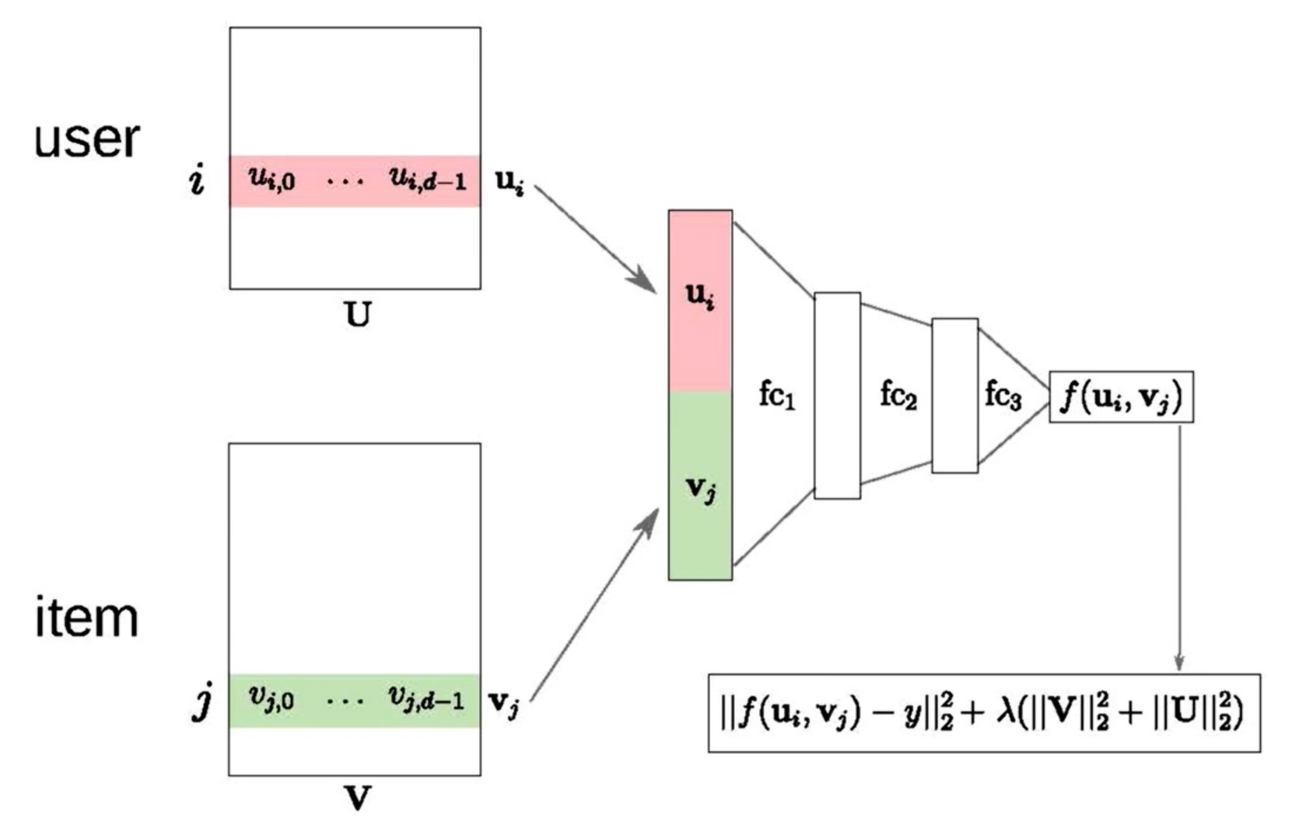
Solution. One-hot representation

- Highly sparse vector
- Each axis has a meaning
- Symbols are equidistance from each other (does not capture the variation)

A Neural network model architecture

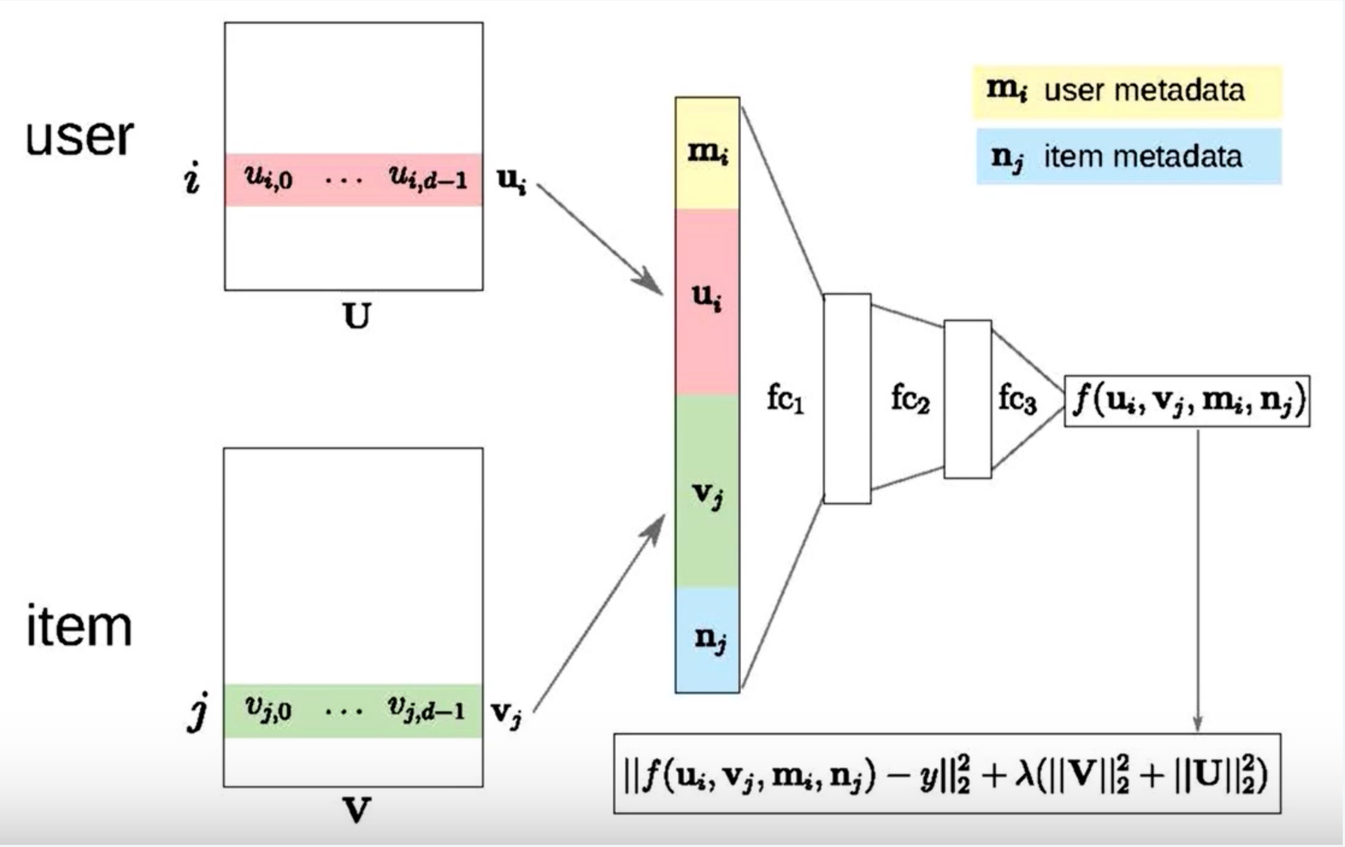
Create a user-item vector and feed into a multi-layer NN.

Training the NN is a supervised learning problem



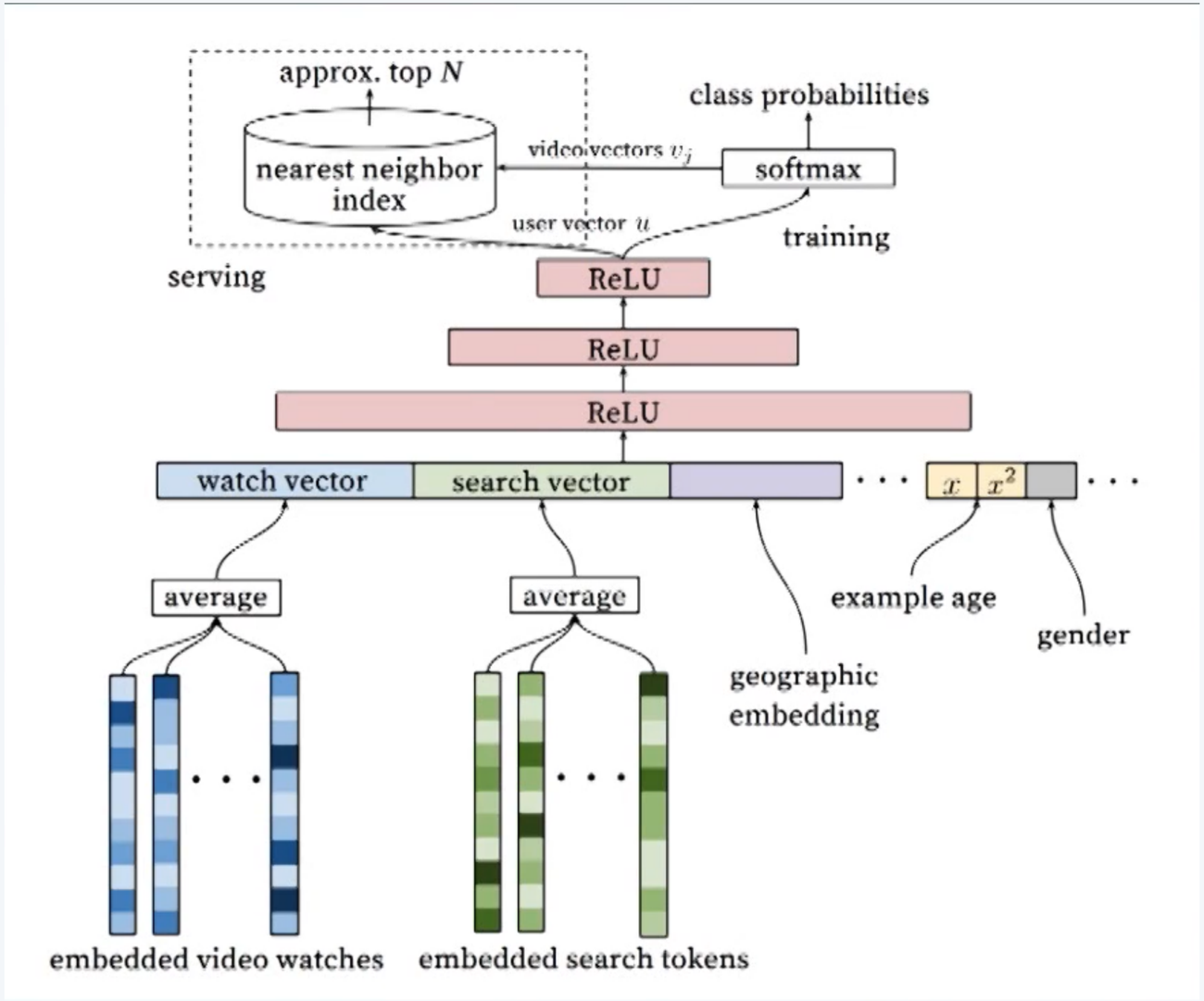
Expanded NN model

In the expanded model, we can also add user metadata to input vector so that more weights can be learned.



YouTube architecture for recommendations

A simplified model of a YouTube recommender



https://research.google.com/pubs/pub45530.html

INTRODUCTION TO DATA SCIENCE

Rutgers University

14.2 Recommender Systems II

https://rutgers.instructure.com/courses/30716

Slide credits> CMU Data Sconce