Recommender Systems || Lecture 3

Summer 2022

Dr. Karthik Mohan

Univ. of Washington, Seattle

June 30 2022

Last Lecture

1. Recommender baseline based on Cosine Similarity
2. Breakout discussions on real use cases
3. Distance vs similarity metrics
4. Behavioral vs Content based recommendations

Today's Lecture

1. Bag of words model for content based recommendations
2. SVD and Matrix Factorization methods
3. Twitter recommendations

Week by Week Break Down (Tentative)

Week	Lecture Material	Assignment
1	Intro to Recommender Systems	Sambazon case study
2	Recommender System Baselines	Shopify case study
3	Matrix Factorization methods	Twitter case study
4	Matrix Factorization methods	Twitter case study
5	Deep Learning based recommendations	Walmart case study
6	ML Pipeline for Recommender systems	Amazon case study
7	Real-time Recommendations	Amazon Fresh case study
8	Diversity and Relevance	Final Project
9	Scaling Recommender systems	Final Project
10	Special Topics	Final Project

Simple Baseline Recommender

Cosine Similarity

Let's say you want to recommend products from some categories (e.g. groceries, music and fitness to customers) in a way that is personalized and so they find it relevant. Here's a simple baseline model:

1. **Representation:** Find a way to represent your product by a fixed length vector of some dimension d. d here usually captures the number of relevant "features".

Simple Baseline Recommender

Cosine Similarity

Let's say you want to recommend products from some categories (e.g. groceries, music and fitness to customers) in a way that is personalized and so they find it relevant. Here's a simple baseline model:

1. **Representation:** Find a way to represent your product by a fixed length vector of some dimension d. d here usually captures the number of relevant "features".
2. **Cosine Similarity:** Let's say you have a reference product p and candidate products to recommend $p_1, p_2, \ldots$. Evaluate the cosine similarity between product p and all the other products.

Simple Baseline Recommender

Cosine Similarity

Let's say you want to recommend products from some categories (e.g. groceries, music and fitness to customers) in a way that is personalized and so they find it relevant. Here's a simple baseline model:

1. **Representation:** Find a way to represent your product by a fixed length vector of some dimension d. d here usually captures the number of relevant "features".
2. **Cosine Similarity:** Let's say you have a reference product p and candidate products to recommend $p_1, p_2, \ldots$. Evaluate the cosine similarity between product p and all the other products.
3. **Rank:** Rank the candidates for recommendation based on cosine simialrity scores (highest to lowest).

Simple Baseline Recommender

Cosine Similarity

Let's say you want to recommend products from some categories (e.g. groceries, music and fitness to customers) in a way that is personalized and so they find it relevant. Here's a simple baseline model:

1. **Representation:** Find a way to represent your product by a fixed length vector of some dimension d. d here usually captures the number of relevant "features".
2. **Cosine Similarity:** Let's say you have a reference product p and candidate products to recommend $p_1, p_2, \ldots$. Evaluate the cosine similarity between product p and all the other products.
3. **Rank:** Rank the candidates for recommendation based on cosine similalrity scores (highest to lowest).
4. **Truncate:** Pick top k from the ranked list to recommend. Where k is usually decided by specifications of your recommender widget.

Behavioral vs Content Based Recommendations

1. **Behavioral:** Recommendations on understanding customer behaviors based on data (purchases, views and other data). **Collaborative Filtering** is a modeling approach in recommendations that is solely dependent on behavioral data. *Customers who bought this also bought this..* is an example of a behavioral recommender.
2. **Content:** Recommenders that use content information on items or products are called content based recommenders. *Items similar to what you purhcased* ... is an example of this.
3. **Hybrid:** Often times, a mixture of behavioral and content based data leads to better overall recommender widgets.

Content based recommendations

1. Use features (or content) from the products to make recommendations.

Content based recommendations

1. Use features (or content) from the products to make recommendations.
2. This includes product categories and product brand

Content based recommendations

1. Use features (or content) from the products to make recommendations.
2. This includes product categories and product brand
3. Anything else?

Breakout discussion (from previous lecture)

content based recs → makes sense
Behavioral recs → ✗ (no data)

New product on the block

Your team is launching new products and product categories to the massive product line of Sambazon. You are the new data scientist hired to surface these new products to customers so your team can drive the sales for the launched products. As of now, there is no easy way for a customer to see the new products, as it isn't yet popular and also because the new products have very little sales to be surfaced by other widgets that exist on the Sambazon recommendations pipeline. How would you leverage what you know about the new product line including any product descriptions and meta data to drive up the sales for the products. Assume the new products are in a product category that a significant percentage of Sambazon customers have purchased from in the past. Come up with a clear step-by-step model/algorithm/approach and also discuss how you will evaluate your recommender model?

Content Recommendations Baseline: Bag of Words Representation

Bag of Words

When we have a dictionary of words and any sentence represented as a binary vector of the dictionary length, where a 1 represents a word present in the sentence and 0 if not - This is a bag of words representation.

Content Recommendations Baseline: Bag of Words Representation

Bag of Words

When we have a dictionary of words and any sentence represented as a binary vector of the dictionary length, where a 1 represents a word present in the sentence and 0 if not - This is a bag of words representation.

Bag of Words Example

Let the dictionary of words be $\{hello, bye, later, today\}$ and a sentence be "Hello, how are you doing today?" What would be the bag of words representation for the sentence (minus the punctuations)?

Sentence 1:- "Hello, how are you doing today?"

Sentence 2:- "I have to go now. Bye!"

hello	bye	later	today	
1	0	0	1	→ $\in \mathbb{R}^4$ 4 dim BoW vector ↳ BoW for S1!
0	1	0	0	↳ BoW for S2!

ICE #1

Bag of words for content representation

Let a dictionary consist of the following words: dictionary
$\{$*fitness*, *women*, *sports*, *clothing*, *clothes*, *bags*, *wear*$\}$.
Assume there is a new Sambazon product that just came out today with the following one line description: "The *Avo fitness wear series 9* brings to you fitness clothing exclusively for women athletes and health conscious women looking to up their game." The dimension of the bag of words representation vector and the number of non-zero elements in the representation are respectively:

1. 6 and 3
2. 6 and 4
3. 7 and 4
4. 8 and 3

(This Lecture ends here!)

Breakout discussion #1

Bag of words (BOW) for behavioral representation

Let's say we have purchase history data on customers - Specifically which product was purchased when by which customer. Assume we do not have any content data available on the products (e.g. description, brand, category, etc). We still want to represent each customer by a vector. Think of a bag of words (BOW) representation that can give a fixed length vector representation for each customer, specifically based on behavioral data. What would be the pros/cons of your representation?

Breakout discussion #2

BOW for Fashion

Let's say you have fashion products line where what's fashionable last year may not be fashionable today. You still want to leverage past purchase data to recommend fashion clothing to customers. Given that you have the purchase and view history data of fashion products for each customer for the past one year - Can you think of an effective BOW representation to represent a customer? What would be the length of the representation? How about when we want to have an effective BOW representation for a product - How would you design this? How do you know a representation is effective or not?

BOW summary

1. BOW gives a starter representation for any model

BOW summary

1. BOW gives a starter representation for any model
2. BOW data representation enables us to get started with baseline recommenders both for content and behavioral models

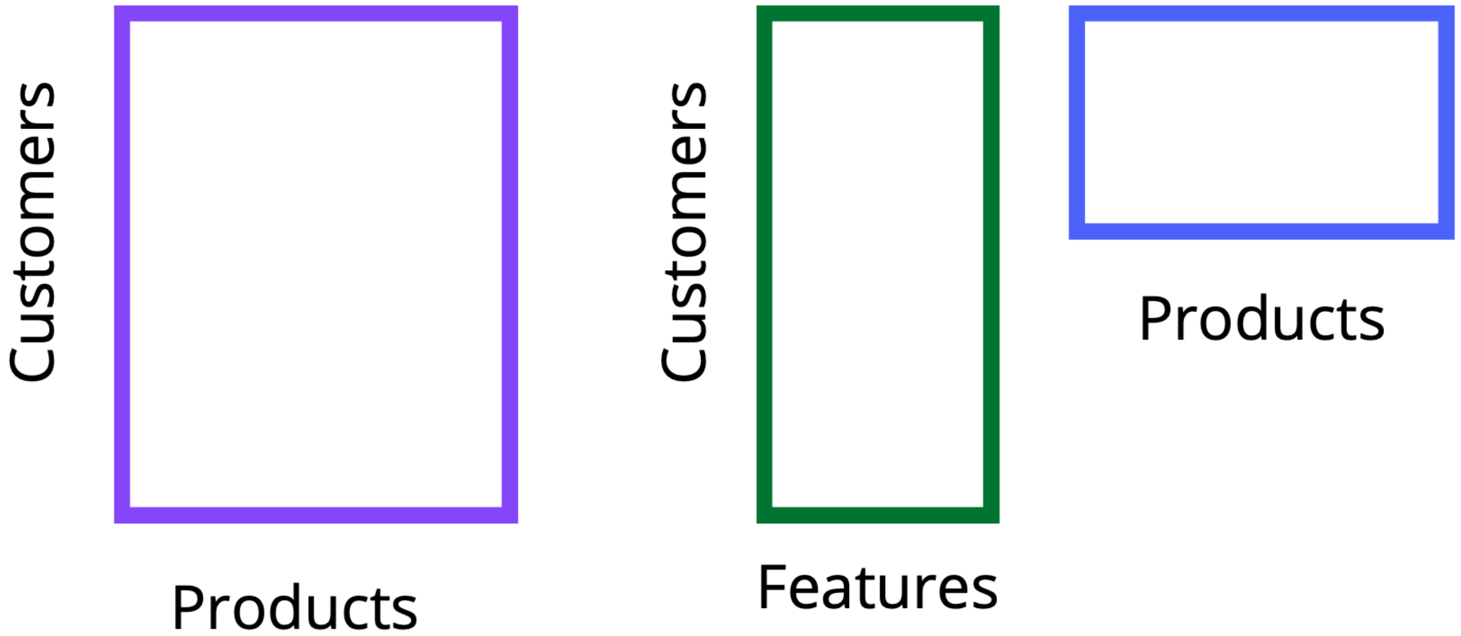
BOW summary

1. BOW gives a starter representation for any model
2. BOW data representation enables us to get started with baseline recommenders both for content and behavioral models
3. For many data sets, BOW can lead to heavily sparse representations (lots of zeros) making the representation unwieldy

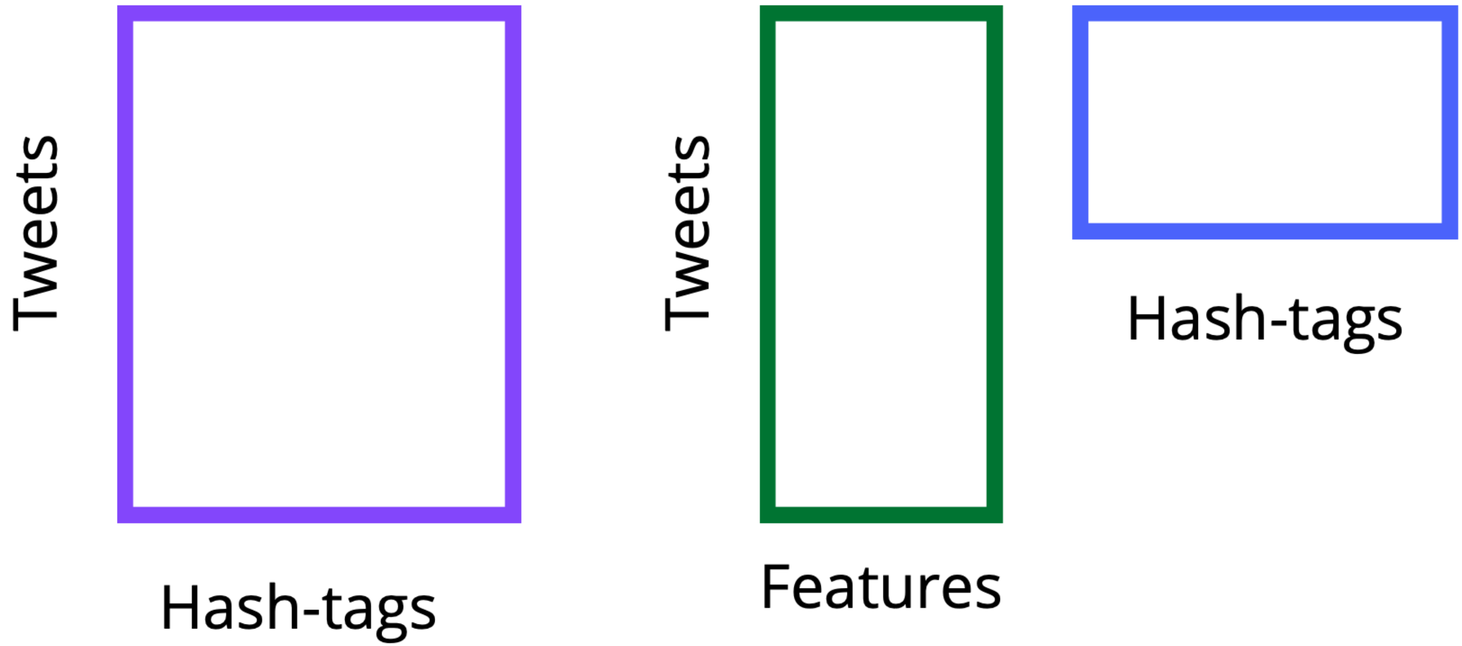
BOW summary

1. BOW gives a starter representation for any model
2. BOW data representation enables us to get started with baseline recommenders both for content and behavioral models
3. For many data sets, BOW can lead to heavily sparse representations (lots of zeros) making the representation unwieldy
4. In Deep Learning models, even if BOW is a starter representation, the DL models learn more refined, data driven, learned and compact representations for both customers and products (e.g. 256 dim instead of 100k dim representation)

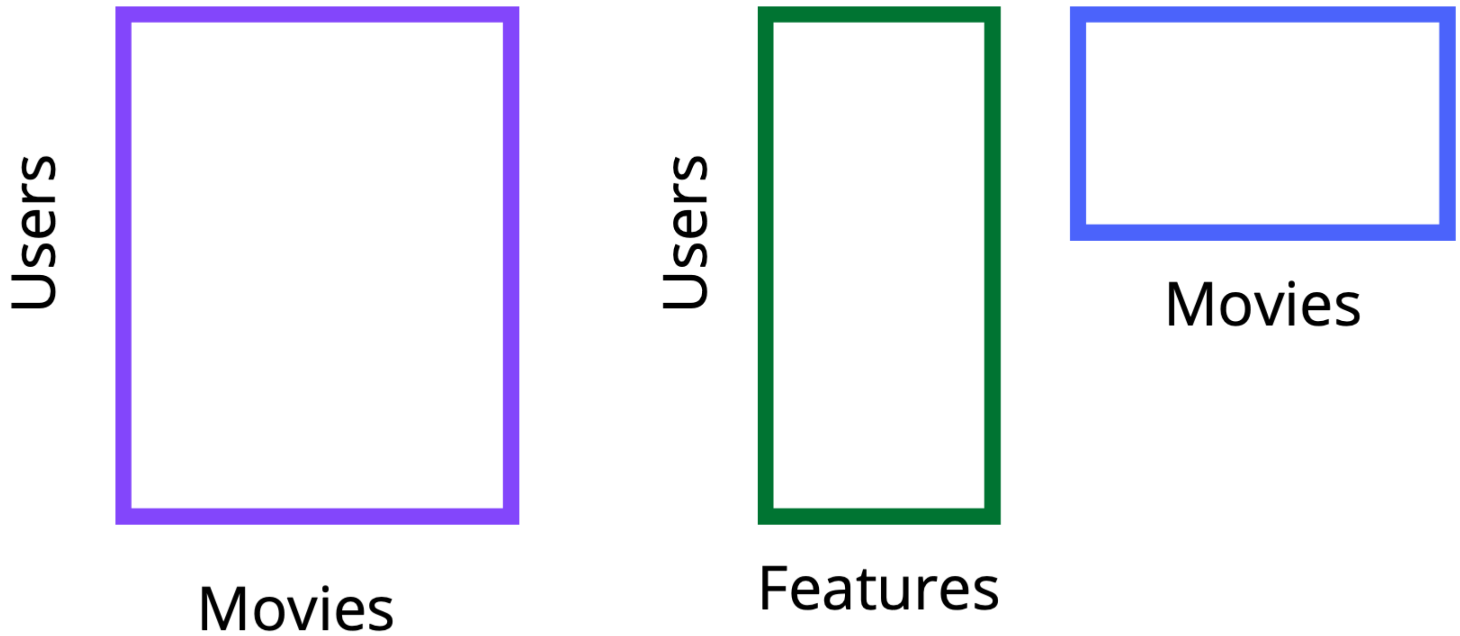
Matrix Factorization: Refined Behavioral Recommendation Models



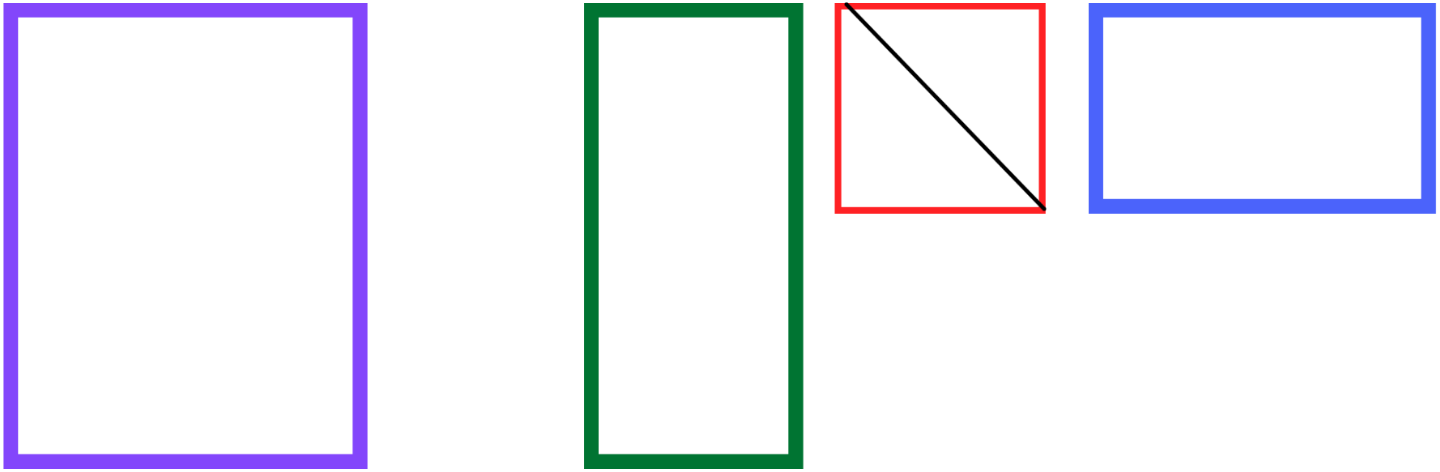
Matrix Factorization: Refined Behavioral Recommendation Models



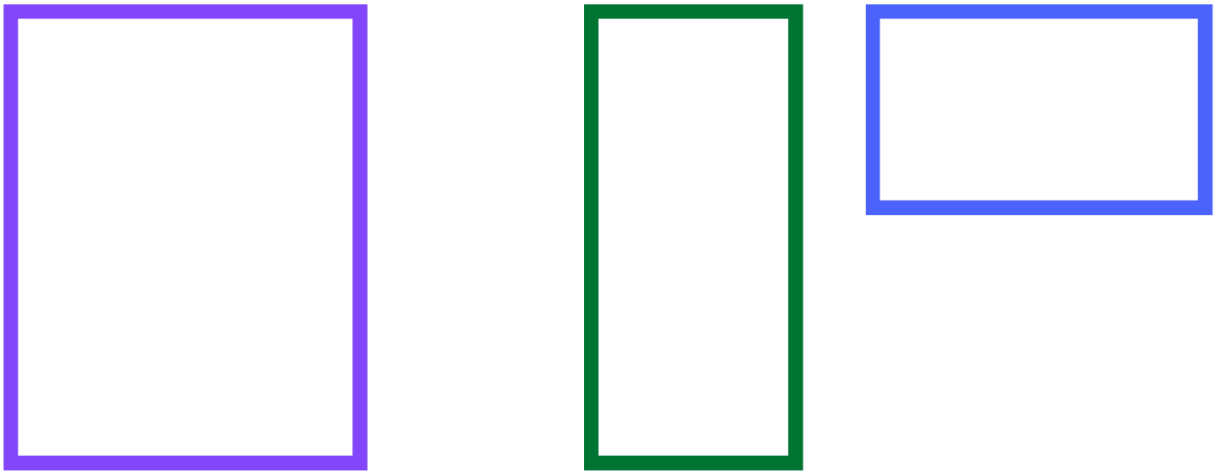
Matrix Factorization: Refined Behavioral Recommendation Models



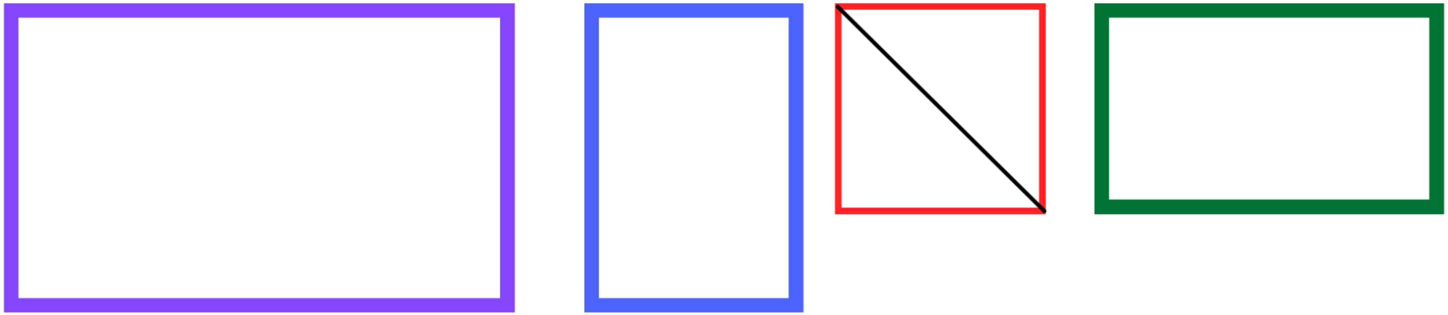
Matrix Factorization: SVD - Baseline Model



Matrix Factorization: SVD - Baseline Model

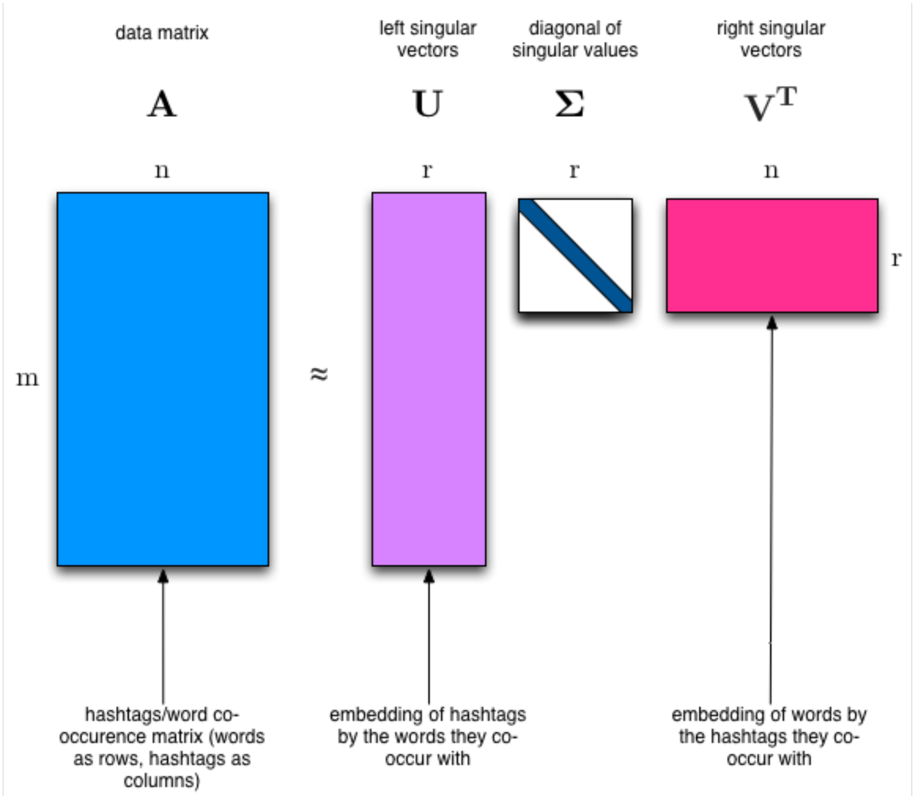


Matrix Factorization: SVD - Baseline Model



SVD vs Eigen Decomposition

Matrix Factorization: SVD for Tweet embeddings and recommendations



Similarity Search using Embeddings

Breakout discussion #3

Assume that you would like to build a tweet recommender system to your customer base. You have two models for recommendations: a) First model does similarity search on the "raw customer-tweet views matrix". I.e. for any tweet a customer has viewed, you search for similar tweets based on cosine similarity using the raw BOW representation of the tweet. Assume the matrix on an average has 1000 non-zeros per row and 10000 non-zeros per column
b) You use the factors coming off of a SVD model with 250 embedded dimensions to do similarity search By what factor would method b) be faster on an average than method a) assuming you were finding top 10 tweets to recommend for each of a thousand reference tweets. Also assume speed is measured based on number of computational flops (as compared to an actual code run comparison).

Similarity Search using Embeddings (continued)

a) 30
b) 40
c) 50
d) 100

BOW vs Learned Representations

Breakout discussion #4

Handling Missing Values

When we have a matrix of customers and their product purchases - It's straightforward to apply SVD and get customer embeddings and product embeddings from the factored matrices. This enables us to search for similar products given a reference product, or similar customers given a reference customer. However, there is a catch here. The number of products in a big retail company may number in hundreds of thousands and the number of customers might number in millions. So we are looking at $1MM \times 100k$ matrix or larger. Do we have a value for every cell in this matrix? The answer is no, as customers only buy a tiny selection of the catalog. And conversely, a new product may not have enough purchases. If a matrix is incomplete, how would we perform an SVD to get factors? Discuss pros and cons of your approaches in this case.

Missing Value Imputations

Next Time

- Better algorithm than SVD - Matrix Factorization

Next Time

- Better algorithm than SVD - Matrix Factorization
- New use case

Next Time

- Better algorithm than SVD - Matrix Factorization
- New use case
- Assignment 2 will be assigned Friday/Saturday and due next week

Next Time

- Better algorithm than SVD - Matrix Factorization
- New use case
- Assignment 2 will be assigned Friday/Saturday and due next week
- Enjoy the long weekend!

Breakout discussion #5

Repeat purchase recommendations

Assume you are working in a team at Sambazon that is building a repeat purchase recommender system. You have access to anonymized data on customers past purchase history over the past year for popular categories of products. What would be a simple baseline model for repeat purchase recommendations? What factors would you consider when you design your baseline model (your factors have to tie in with the bottom line/top line metrics that your team/project cares about) How would you evaluate the goodness of your model? What specific quantitative metrics would you use to measure the goodness? What are the pros/cons of your baseline and how would you improve it?