Developing a Coupled Matrix-Tensor Factorization Model in Python to Analyze Mobility Data for Disaster Response

Lisa Fung

Introduction

Real-world Issue

- To **improve evacuation routes and disaster response** in California, we must understand the **underlying factors of human movement** using mobility data.

Computational Problem

- Difficult to **combine mobility datasets** with different measured variables and find **shared hidden factors** in mobility data

Coupled Matrix-Tensor Factorization (CMTF)

- Data fusion and tensor decomposition algorithm that **merges data** and **simultaneously decomposes** data coupled in tensors and matrices

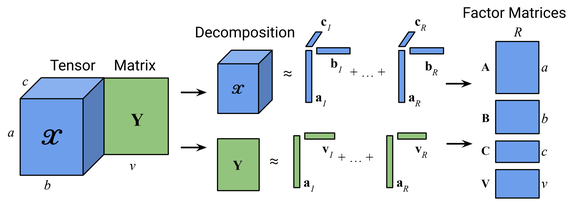


FIG. 1. Coupled matrix-tensor factorization of tensor $\mathcal{X}$ and matrix $\mathbf{Y}$ coupled along the vertical axis a. The tensor $\mathcal{X}$ and matrix $\mathbf{Y}$ are approximated by a sum of outer-multiplied component vectors represented as factor matrices.

CMTF Implementations

Alternating Least Squares Optimization

- Optimizes **one factor matrix at a time** while keeping other factor matrices constant
- TensorLy library in Python, Systems serology implementation
  - **Fast decomposition**, easy to implement

Gradient-based Optimization

- Finds **minimum of loss function** using gradients to optimize factor matrices
- CMTF Toolbox in MATLAB, S3CMTF in C++ using parallelized stochastic gradient descent
  - Takes longer to run, **achieves better accuracy**

Engineering Goal

Develop a CMTF software (CMTF-OPT1) using gradient-based optimization in Python

- Using synthetic data, quantify **speed and accuracy** of gradient-based CMTF model against TensorLy's CMTF-ALS

Present an application of CMTF to study **mobility in the Greater Los Angeles area**

- Utilize Core Consistency Diagnostic (CORCONDIA) and Factor Match Score (FMS) to determine **optimal CMTF decomposition rank** (NP-hard problem)
- Interpret **latent factors affecting human movement** from factor matrices, using geographic maps and time series plots

Coupled Matrix-Tensor Factorization Software: CMTF-OPT1

CMTF Initialization

Conjugate Gradient Optimization

Output Factor Matrices

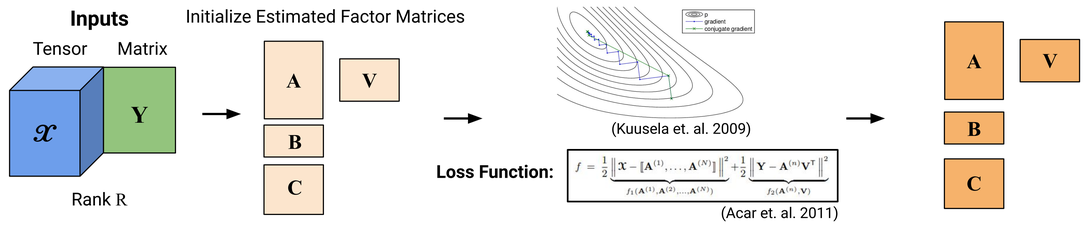
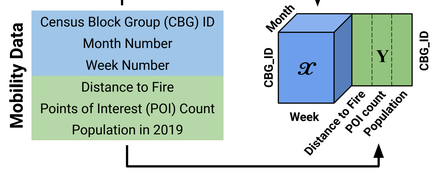


FIG. 2. CMTF-OPT1 takes in a **tensor, matrix, and rank R** as inputs, as well as other optimization parameters. Then, factor matrices **A**, **B**, **C**, and **V** are initialized using singular value decomposition (SVD) or random initialization. From there, **nonlinear conjugate gradient optimization** is used to improve the estimated factor matrices by **minimizing the non-convex loss function using gradients**. Finally, the optimized factor matrices **A**, **B**, **C**, and **V** are returned.

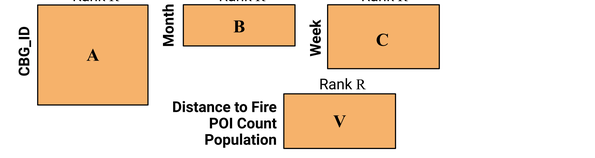
- After comparing CMTF-OPT1 to TensorLy CMTF-ALS, I decided to use **TensorLy CMTF-ALS in Mobility Data Analysis** because of its fast runtime in decomposing large tensors and matrices (see Results).

CMTF Application: Mobility in the Greater Los Angeles area

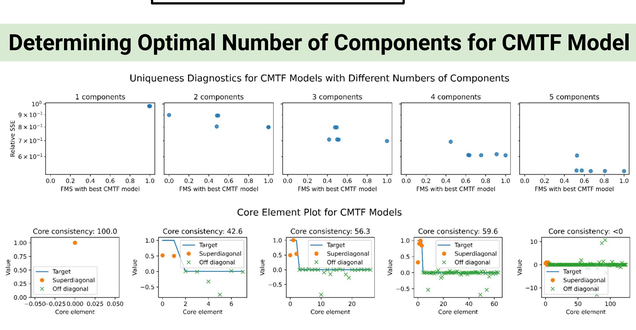
Mobility Data Stored in Tensor and Matrix

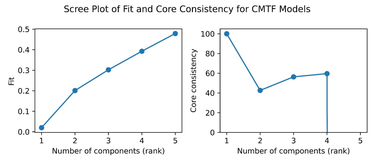


Factor Matrices: Hidden Factors in Human Mobility



Determining Optimal Number of Components for CMTF Model



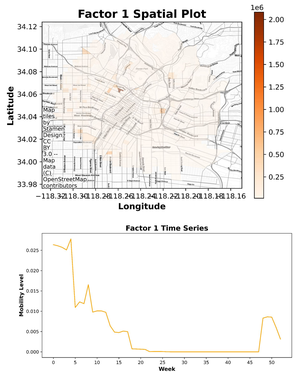


Diagnostics indicate **Rank-4 CMTF model** is best:

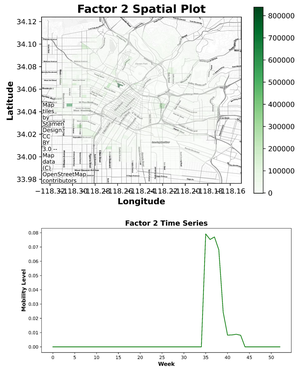
- **Factor matrices are similar** as determined by Factor Match Score (FMS) and low Sum of Squares Error (SSE)
- **Core approximated tensor is similar to original tensor** as shown in Core Element Plot
- **Rank-4 model fits multilinear behavior** as seen by sharp drop at 4 components in Core Consistency Plot

Interpreting Hidden Factors of Human Mobility Using Rank-4 CMTF Model

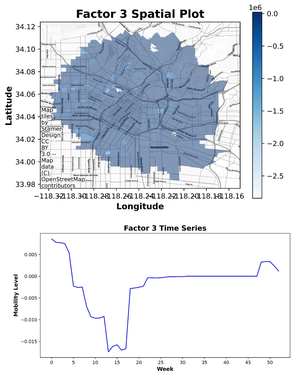
Factor 1: Natural Parks and Recreation



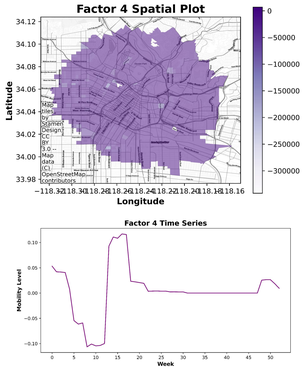
Factor 2: Shopping Areas



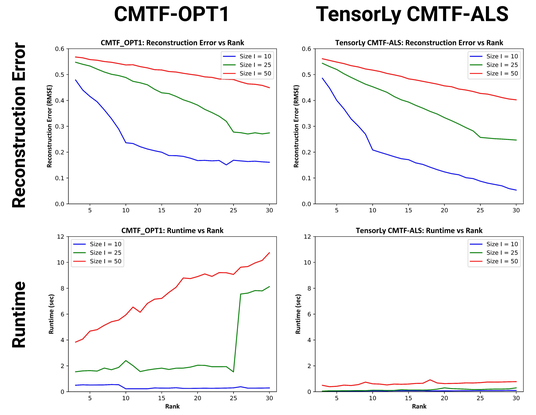
Factor 3: Housing Areas



Factor 4: Housing Areas



Synthetic Data Comparison



Setup

- CMTF decomposition of I x I x I tensor and I x I matrix using CMTF-OPT1 and TensorLy CMTF-ALS
- Ran 5 trials per rank = 3-30. Graphed average RMSE, runtime

Performance

- **CMTF-OPT1 performs on par** with TensorLy CMTF-ALS in terms of **accuracy** (measured by reconstruction error): Both had **RMSE under 0.57**
- CMTF-OPT1 has a much **longer runtime** than TensorLy CMTF-ALS

Conclusions

CMTF-OPT1 Software Development

- Promising use of gradient-based optimization in Coupled Matrix-Tensor Factorization for accuracy

Mobility Data Analysis using CMTF

- **Effective in identifying hidden factors** in mobility data using spatial and temporal interpretation
- **Streamlined workflow** for identifying optimal rank and performing CMTF decomposition on mobility data

Future Directions

- Test **alternative gradient-based optimization** techniques in CMTF-OPT1 for faster speed
- Adapt TensorLy CMTF-ALS for **sparse tensors and matrices** to efficiently decompose real-world data

Future Application

- Cities can collect mobility data through different sources and **utilize CMTF-OPT1 to uncover hidden factors** in human mobility. This can be used to **simulate human behavior** in wildfire simulations for cities to identify the **best evacuation routes**.

Selected References

1. E. Acar, T. Kolda and D. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," *Proc. Mining and Learning with Graphs (MLG'11)*, Aug. 2011.
2. J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "TensorLy: Tensor Learning in Python," *Journal of Machine Learning Research*, vol. 20, no. 26, Feb. 2019.
3. "Places data curated for Accurate Geospatial Analytics," *SafeGraph*, 2022. [Online]. Available: https://www.safegraph.com/.