Comparing Different Matrix Factorization Techniques for Unsupervised Learning

Qijia Jiang[1], Boying Meng[1] and Xuyang Lu[1]

[1]*Department of Electrical and Computer Engineering, Rice University, Houston, Texas, U.S.A*

Introduction

In real life, often, we come across data that doesn't have outputs (labels). In these cases, we strive to find the hidden structure and latent groupings within the data. This is when unsupervised machine learning comes into play. Popular unsupervised learning techniques include K-means, SOM and various clustering algorithms (spectral clustering, hierarchical clustering). Here, we discuss unsupervised machine learning within the context of various matrix factorization techniques.

Motivation

- Let's assume we have a data matrix of size nxp, where n corresponds to n observations and p corresponds to p features; very often, p is very large and the data is noisy. What we want is to find the association among the variables and uncover the major patterns underlying the data.
- What matrix factorization does is to reduce the dimensionality – It projects data matrix into a lower dimension space so what seems random and messy in the high dimension becomes structured and grouped in lower dimension.

PCA

- Short for Principal Component Analysis – closely related to SVD and eigenvalue problem (Ridge Regression, also)
- First PC (Principal Component) – direction that maximizes the sample variance
- Succeeding ones – direction that gives highest variance under the constraint of it being orthogonal (uncorrelated) to preceding ones
- Model looks like this:

$$\underset{U,D,V}{\text{minimize}} \frac{1}{2}\left\|X - UDV^T\right\|$$

$$\text{subject to } U^TU = I,\quad V^TV = I,\quad D \in diag^+$$

- Interpretation (pattern recognition): first column of U gives the first major pattern in sample (row) space; first column of V gives the first major pattern in feature (column) space
- Essentially, we are doing a coordinate transformation and the newly formed coordinate axes are the major patterns we are looking for
- Really easy to implement: feed the data matrix(n*p) to the SVD command in Matlab, extract the PC loading(V) and PC score(U) vector

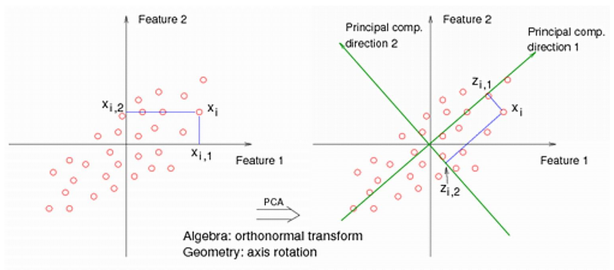


Figure Credit: https://onlinecourses.science.psu.edu/stat857/node/35

ICA

- Short for Independent Component Analysis
- Imagine the data we have is a linear mixture of unknown latent factors and the factors are mutually independent and non-Gaussian, our goal is to identify these components
- Weakness of PCA: assumes Gaussianity – the condition $U^TU = I$ and $V^TV = I$ imply independence only when the data is Gaussian; what ICA does is to find statistically independent (rather than uncorrelated) sources
- Model:

$$X = AS$$

$$\hat{S} = W\tilde{X}$$

  - where A is the mixing matrix and S is the source signals (rows of S independent)
  - We recover the signal using the "whitening" matrix W (where $W = A^{-1}$)
- Algorithms: Entropy-based, Neg-Entropy (Gaussian has maximal entropy, we want it to be as far from Gaussian as possible)
- What we use: FastICA[2] algorithm converges quickly

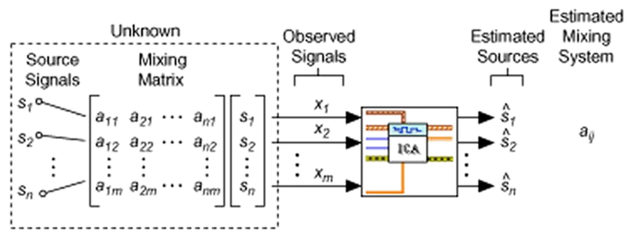


Figure Credit: http://zone.ni.com/reference/en-XX/help/372656B-01/lvasptconcepts/tsa_multivariate_stat_analysis/

NMF

- Short for Nonnegative Matrix Factorization
- Solves another weakness of PCA: PCA tends to group both positively correlated & negatively correlated components together
- Model looks like this (for continuous data):

$$\min_{W,H} \frac{1}{2}\|X - XH\|_2^2$$

$$\text{subject to } W_{ij} \geq 0,\ H_{ij} \geq 0$$

- What NMF does: by forcing W and H to be positive, it finds patterns with same direction of correlation
- Another point of view: fuzzy clustering – W gives probabilistic cluster memberships, columns of H give variables that define each
- What we use: nnmf command in Matlab
- For count data, model looks like this:

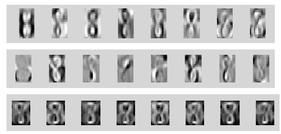
$$\underset{W,H}{\text{maximize}} \sum_{i=1}^{n}\sum_{j=1}^{p}\left[X_{ij}\log\left(W_iH_j\right) - W_iH_j\right]$$

$$\text{subject to } W_{ij} \geq 0,\ H_{ij} \geq 0$$

Result & Comparison

- **Handwritten Digits Dataset[3] (use digit 8 as example)**
  - Some example digits

  - Comparing PCA, ICA and NMF by plotting the first 8 components

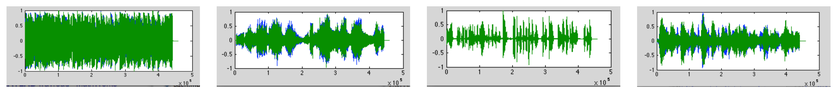


(PCA)

(ICA)

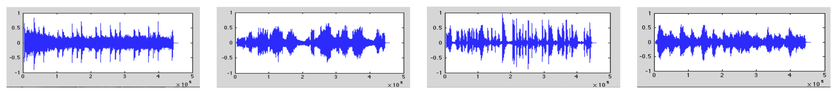
(NMF)

  - For this dataset, we see that NMF performs better than PCA and ICA. It extracts the most significant features of 8.
- **Cocktail Party Problem (Blind Source Separation)**
  - Four different mixtures of four sources: Bach's Symphony, CNN News Material, Finnish Song and Pop Song Breakeven
  - Fed into FastICA algorithm, 8 components in total (two channels for each audio file), pick the largest four components
  - Original Waveforms



Source I | Source II | Source III | Source IV

  - ICA works best in this case, it picks out all four sounds (with a little noise) while NMF and PCA only picks out one.
  - Recovered Waveforms using ICA (best result)



Recovered Source I | Recovered Source II | Recovered Source III | Recovered Source IV

- **Interesting application: we succeed in extracting the background music in a song**

Conclusion

Different matrix factorization techniques work well towards different ends:

- NMF works well for modeling non-negative data such as images
- ICA works well for finding independent sources when the data isn't Gaussian
- PCA has much broader application than ICA and NMF; it is ideal for pattern recognition and dimension reduction

Reference

[1] Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning", Springer, Second Edition

[2] Hyvärinen, A (1999). *Fast and Robust Fixed-point Algorithms for Independent Component Analysis*. IEEE Transactions on Neural Networks, 10(3),626-634.

[3] Normalized Handwritten Digits Dataset, Le Cun et al., 1990, AT&T Research Labs

Contact Info: Jiang (*qj2@rice.edu*), Meng (*bm15@rice.edu*), Lu (*xl27@rice.edu*)