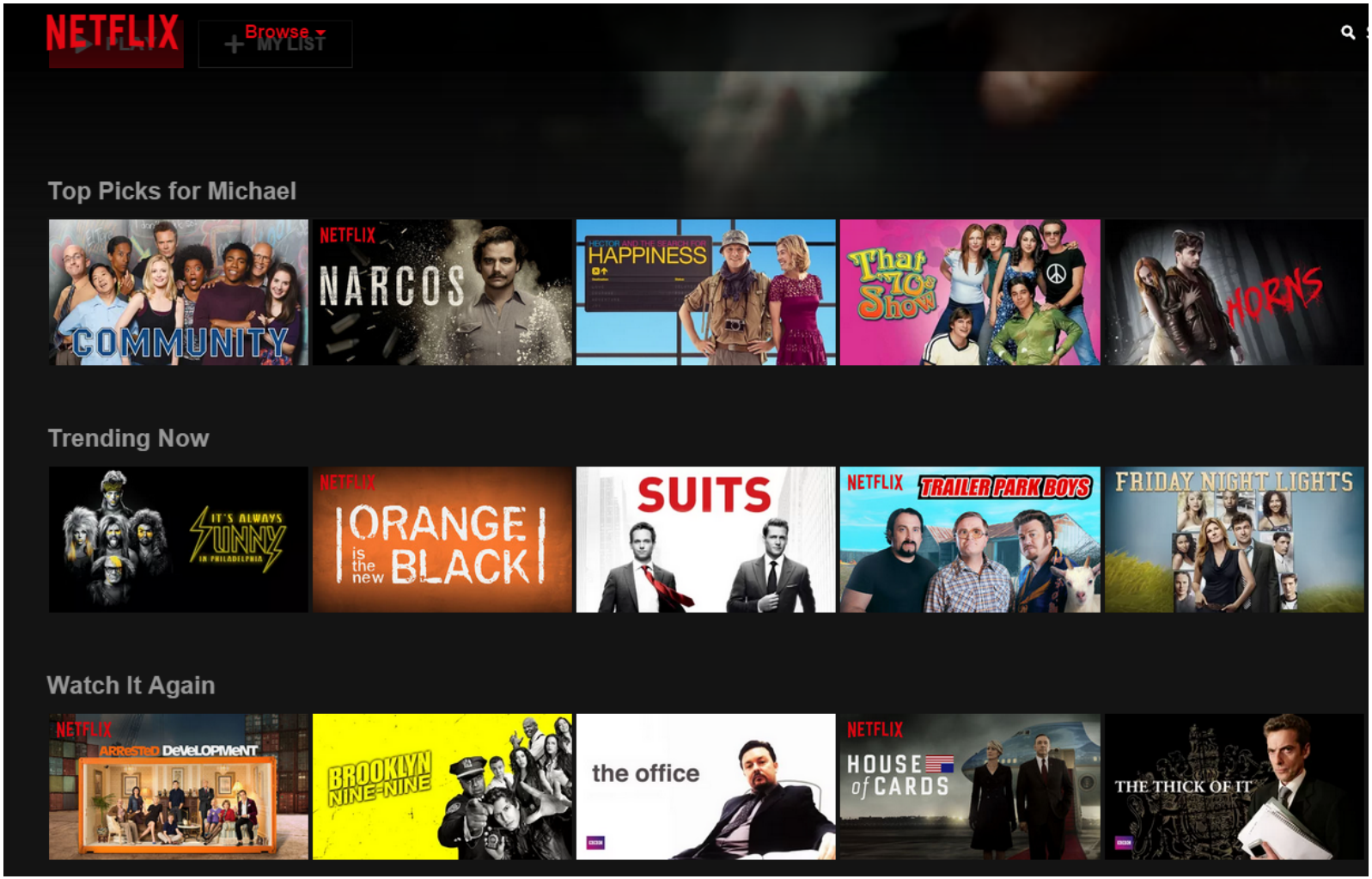
16. Dimensionality reduction

- Recommender systems
- Representations
- PCA
- ICA

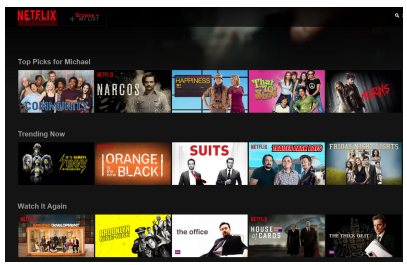
Recommender systems

Recommender Systems



Is this product any good?

- Up until now, we recommend a product based on its features
- e.g. Wizard of Oz any good?
  - cute dog $+1$
  - lots of magic $+1$
  - pretty shoes $+1$
  - cheesy messaging -1



- Predict $y_{\text{like}}(x_{\text{WizOfOz}}) = \mathbf{sign}(\theta^T x_{\text{WizOfOz}})$
- Issue: How to collect so much data?

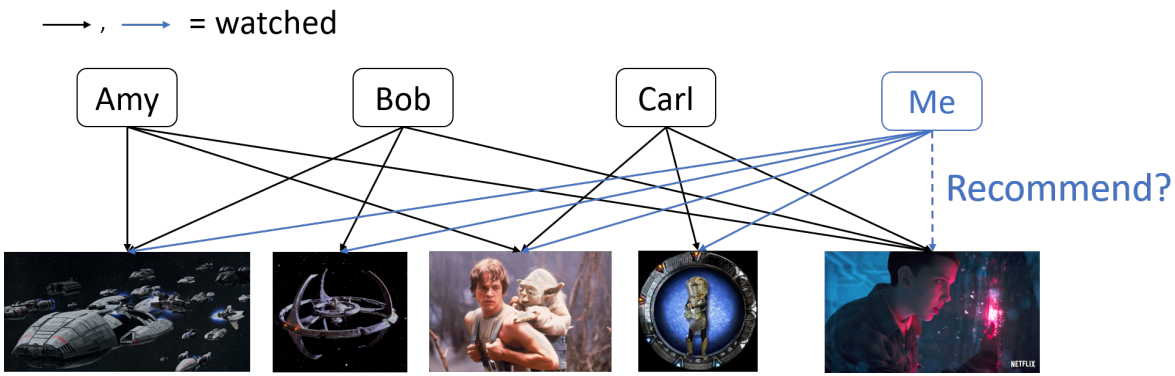
Recommendation without features

- I love SciFi.
- I've seen *Battlestar Galactica*, *Star Trek*, *Star Wars*, *Stargate*, ...




- *Stranger things*?
- It's not "typical SciFi" (no spaceships, aliens).

Recommendation by collaboration



Assumptions:

- There aren't that many types of people ("archetypes", "archetypical")
- There aren't that many types of movies (genres, styles, ...)
- "Does Bob like Star Wars?" hard to learn (I'm not a mind reader!)
- "Does Bob-like people like Star Wars-like movies?" (Tons of data!)

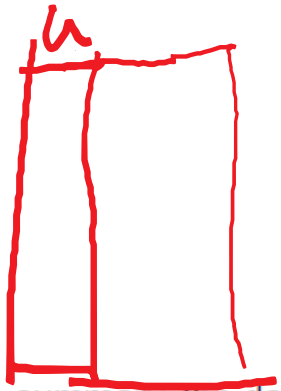
The Netflix Challenge

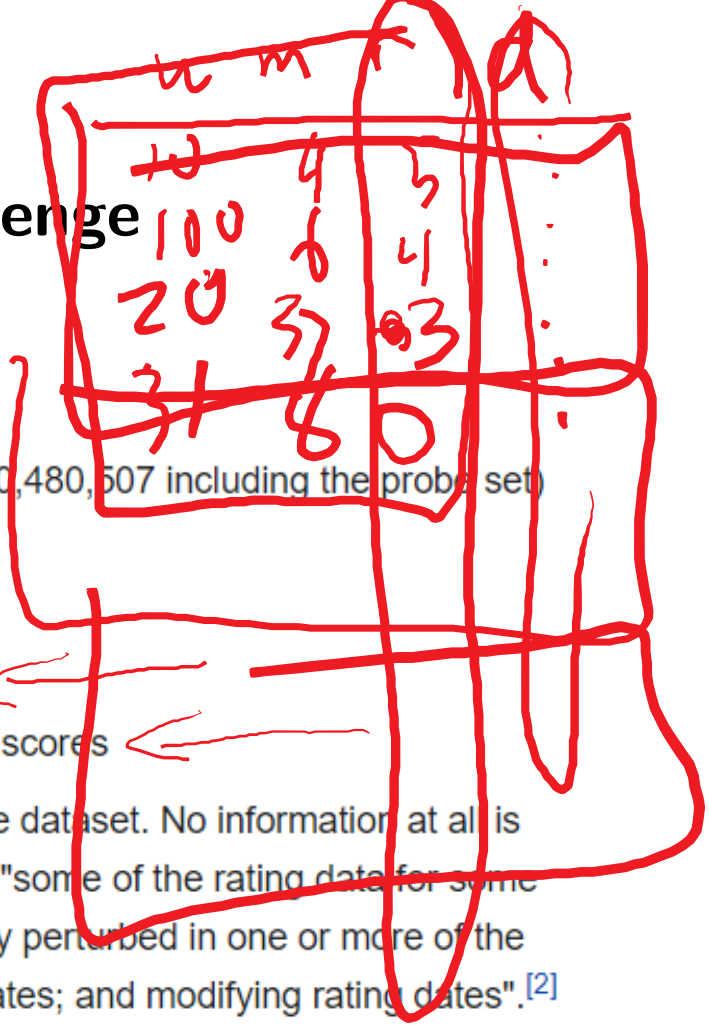
Netflix provided a *training* data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies. Each training rating is a quadruplet of the form `<user, movie, date of grade, grade>`. The user and movie fields are integer IDs, while grades are from 1 to 5 (integral) stars.[3]

The *qualifying* data set contains over 2,817,131 triplets of the form `<user, movie, date of grade>`, with grades known only to the jury. A participating team's algorithm must predict grades on the entire qualifying set, but they are only informed of the score for half of the data, the *quiz* set of 1,408,342 ratings. The other half is the *test* set of 1,408,789, and performance on this is used by the jury to determine potential prize winners. Only the judges know which ratings are in the quiz set, and which are in the test set —this arrangement is intended to make it difficult to hill climb on the test set. Submitted predictions are scored against the true grades in terms of root mean squared error (RMSE), and the goal is to reduce this error as much as possible. Note that while the actual grades are integers in the range 1 to 5, submitted predictions need not be. Netflix also identified a *probe* subset of 1,408,395 ratings within the *training* data set. The *probe*, *quiz*, and *test* data sets were chosen to have similar statistical properties.

(wikipedia)

The Netflix Challenge



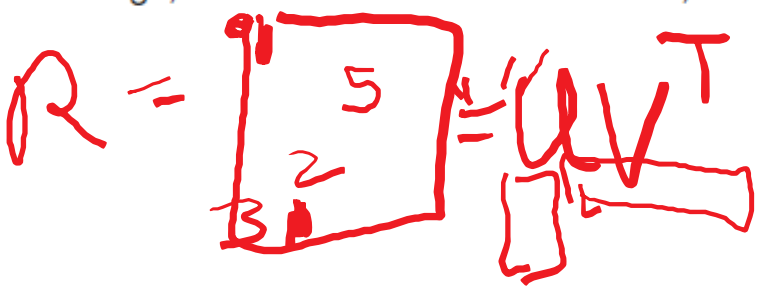


In summary, the data used in the Netflix Prize looks as follows:

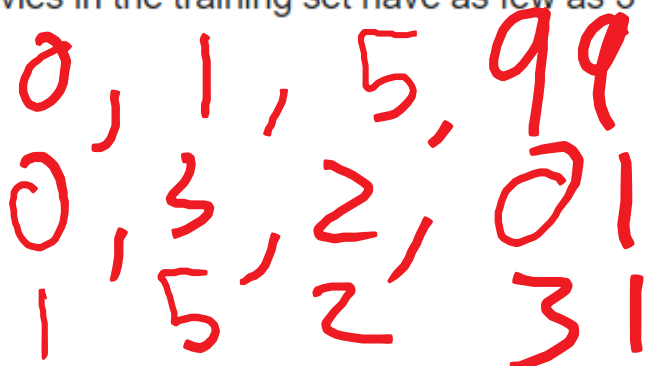
- Training set (99,072,112 ratings not including the probe set, 100,480,507 including the probe set)
  - Probe set (1,408,395 ratings)
- Qualifying set (2,817,131 ratings) consisting of:
  - Test set (1,408,789 ratings), used to determine winners
  - Quiz set (1,408,342 ratings), used to calculate leaderboard scores

For each movie, title and year of release are provided in a separate dataset. No information at all is provided about users. In order to protect the privacy of customers, "some of the rating data for some customers in the training and qualifying sets have been deliberately perturbed in one or more of the following ways: deleting ratings; inserting alternative ratings and dates; and modifying rating dates".[2]

The training set is such that the average user rated over 200 movies, and the average movie was rated by over 5000 users. But there is wide variance in the data—some movies in the training set have as few as 3 ratings,[4] while one user rated over 17,000 movies.[5]



(wikipedia)



The Netflix winner

The BellKor 2008 Solution to the Netflix Prize

Robert M. Bell
AT&T Labs - Research
Florham Park, NJ

Yehuda Koren
Yahoo! Research
Haifa, Israel

Chris Volinsky
AT&T Labs - Research
Florham Park, NJ

BellKor@research.att.com

1. Introduction

Our RMSE=0.8643[2] solution is a linear blend of over 100 results. Some of them are new to this year, whereas many others belong to the set that was reported a year ago in our 2007 Progress Prize report [3]. This report is structured accordingly. In Section 2 we detail methods new to this year. In general, our view is that those newer methods deliver a superior performance compared to the methods we used a year ago. Throughout the description of the methods, we highlight the specific predictors that participated in the final blended solution. Nonetheless, the older methods still play a role in the blend, and thus in Section 3 we list those methods repeated from a year ago. Finally, we conclude with general thoughts in Section 4.

Solution spurred study in collaborative filtering

Matrix factorization

- m users, user i represent as $u_i \in \mathbb{R}^d$
- n movies, movie j represent as $v_j \in \mathbb{R}^d$
- Predicted rating of user i on movie j: R_{ij}
- Form matrices

$$U = [u_1, u_2, ..., u_m], \qquad V = [v_1, v_2, ..., v_n], \qquad R \in \mathbb{R}^{m \times n}$$

- Assumption: $R_{ij} = f(u_i, v_j)$
- Matrix factorization: $R_{ij} = u_i^T v_j$

$$R = UV^T$$

- If $d \ll \min\{m, n\}$, then at most d directions of freedom
- (meta) $\rightarrow$ constrains "how many types" of people, movies

Basic implementation

- Collect data R_{ij} from observed user i, movie j
- Sparse mask: $S_{ij} = 1$ if user i saw movie j, $S_{ij} = 0$ otherwise
- Find u_i, v_j to minimize

$$\underset{u_i \in \mathbb{R}^d, v_j \in \mathbb{R}^d}{\text{minimize}} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij} (R_{ij} - u_i^T v_j)^2$$

- Predict unseen $R_{ij} = u_i^T v_j$
- Idea: by enforcing $d \ll \min\{m, n\}$, we "learn" the unseen R_{ij} via "archetype" information sharing

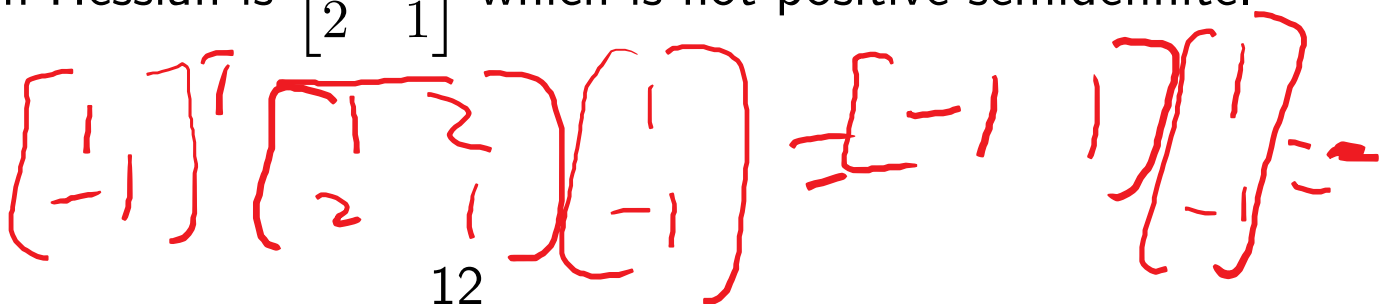
Matrix factorization is a nonconvex problem

$$\underset{u_i \in \mathbb{R}^d, v_j \in \mathbb{R}^d}{\text{minimize}} \ \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{ij} (R_{ij} - u_i^T v_j)^2$$

Proof: Take $d = m = n = 1$, $S = 1$. Then objective is

$$f(u,v) = \tfrac{1}{2}(R - uv)^2, \nabla f(u,v) = (uv - R) \begin{bmatrix} u \\ v \end{bmatrix}, \nabla^2 f(u,v) = \begin{bmatrix} u^2 & 2uv \\ 2uv & v^2 \end{bmatrix}$$

Take $u = v = 1$, then Hessian is $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ which is not positive semidefinite.



Better-than-basic: De-biasing

- Scalar R_0 = average rating for everything
- Some people are just critics: Vector $u_0 \in \mathbb{R}^m$ = average user rating (after $-R_0$)
- Some movies just suck: Vector $v_0 \in \mathbb{R}^n$ = average movie rating (after $-R_0$

New objective function

$$\underset{u_i \in \mathbb{R}^d, v_j \in \mathbb{R}^d}{\text{minimize}} \sum_{i,j:S_{ij}=1} \underbrace{((R_{ij} - R_0 - u_0[i] - v_0[j])}_{\text{de-biased ratings}} - u_i^T v_j)^2$$

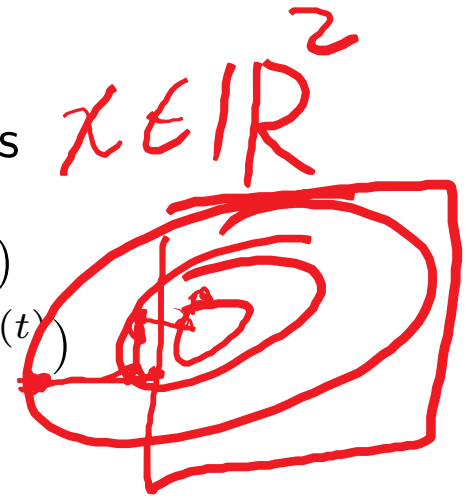
pred : $(u_i^T v_j + u_0[i] + v_0[j] + R_0)$

Implementation of better-than-basic method

$$\underset{u_i \in \mathbb{R}^d, v_j \in \mathbb{R}^d}{\text{minimize}} \; f(U,V) := \sum_{i,j:S_{ij}=0} (\underbrace{(R_{ij} - R_0 - u_0[i] - v_j[0])}_{\text{de-biased ratings}} - u_i^T v_j)^2$$

- Alternating gradient descent: Start with average ratings

$$\begin{aligned} U^{(t+1)} &= U^{(t)} - \alpha \nabla_U f(U^{(t)}, V^{(t)}) \\ V^{(t+1)} &= V^{(t)} - \alpha \nabla_V f(U^{(t+1)}, V^{(t)}) \end{aligned}$$



- Trial and error to pick best α
- Personal experience: about 5-10 passes through data is enough

Further details: `https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf`

RMSE scores on Movielens 100K

Baseline Methods

Software	Method	5-fold CV	all-but-10	References
MyMediaLite 3.07	GlobalAverage	1.1256	1.1238	
MyMediaLite 3.07	UserAverage	1.0437	1.0518	
MyMediaLite 3.07	ItemAverage	1.0246	1.0453	
MyMediaLite 3.07	UserItemBaseline	0.9413	0.9656	

kNN-based Collaborative Filtering

Software	Method	5-fold CV	all-but-10	References
MyMediaLite 3.07	UserKNN	0.9283	0.9572	
MyMediaLite 3.07	ItemKNN	0.9182	0.9445	

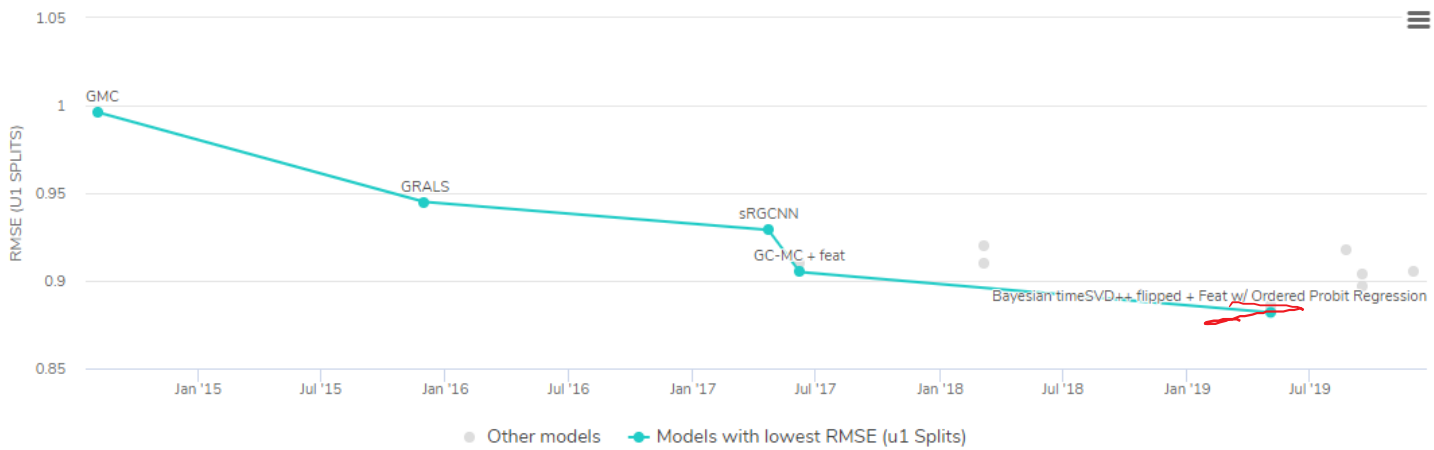
Matrix Factorization

Software	Method	5-fold CV	all-but-10	References
MyMediaLite 3.07	BiasedMatrixFactorization	0.9220	0.9475	
MyMediaLite 3.07	SVDPlusPlus	0.9112	0.9409	
MyMediaLite 3.07	SigmoidUserAsymmetricFactorModel	0.8939	0.9232	

RMSE scores on Movielens 100K

Browse > Miscellaneous > Recommendation Systems > MovieLens 100K dataset

Recommendation Systems on MovieLens 100K



https://paperswithcode.com/sota/collaborative-filtering-on-movielens-100k

RMSE scores on Movielens 100K

RANK	MODEL	RMSE (U1 SPLITS) ↓	RMSE (RANDOM 90/10 SPLITS)	RMSE@80%TRAIN	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	Bayesian timeSVD++ flipped + Feat w/ Ordered Probit Regression	0.882			✓	On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
2	Bayesian timeSVD++ flipped + Feat	0.884			✓	On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
3	Bayesian timeSVD++ flipped	0.886			✓	On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
4	GraphRec + Feat	0.897	0.883		✓	Attribute-aware non-linear co-embeddings of graph features			2019
5	GraphRec	0.904	0.887		×	Attribute-aware non-linear co-embeddings of graph features			2019
6	IGMC	0.905			×	Inductive Matrix Completion Based on Graph Neural Networks			2019
7	GC-MC + feat	0.905			✓	Graph Convolutional Matrix Completion			2017
8	GC-MC	0.910			×	Graph Convolutional Matrix Completion			2017
9	Self-Supervised Exchangeable Model	0.91			×	Deep Models of Interactions Across Sets			2018
10	GRAEM	0.9174			✓	Scalable Probabilistic Matrix Factorization with Graph-Based Priors			2019

https://paperswithcode.com/sota/collaborative-filtering-on-movielens-100k

Ongoing research!

- If $R_{ij} \in \{$  $\}$, replace mean squared with logistic loss
- If $R_{ij} = 1$ if watched, 0 if not watched, need something more creative
- Side information: we shouldn't just ignore features, if we do have them!
- Sharing accounts, how to disambiguate?
- Incorporate movie reviews? NLP?

Representations

Sentiment	Tweets
Negative	@united is the worst. Nonrefundable First class tickets? Oh because when you select Global/FC their system auto selects economy w/upgrade.
	@united I will not be flying you again
Neutral	@VirginAmerica my drivers license is expired by a little over a month. Can I fly Friday morning using my expired license?
	@VirginAmerica any plans to start flying direct from DAL to LAS?
Positive	@VirginAmerica done! Thank you for the quick response, apparently faster than sitting on hold ;)
	@united I appreciate your efforts getting me home!

`ipullrank.com/step-step-twitter-sentiment-analysis-visualizing-united-airlines-pr-crisis/`

Feature extraction?

- Count # good / bad words (works well, easy to trick)
- Count # good / bad known phrases (too many combinations!)
- How to turn a non-numeric feature into a number-like feature?

Co-occurance = similarity

the quick brown fox jumped over the brown fence

- ID words

  1 = the, 2 = quick, 3 = brown, 4 = fox, 5 = jumped, 6 = over, 7 = fence

- Frequency vector:

$$f = (2, 1, 2, 1, 1, 1, 1)$$

- Co-occurance matrix, sliding window = 3

$$C = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Word embedding: $C_{ij} \approx u_i^T u_j$

Word embeddings

Example: vector = histogram of cooccuring words (Lund, Burgess '96)

I have a big fluffy gray cat named Fluffy. We saw a red balloon fly by.

$$x_{\text{cat}} = [\underbrace{0}_{\text{apple}}, \underbrace{0}_{\text{balloon}}, \underbrace{2}_{\text{fluffy}}, \underbrace{1}_{\text{gray}}]$$

$$x_{\text{red}} = [\underbrace{0}_{\text{apple}}, \underbrace{1}_{\text{balloon}}, \underbrace{0}_{\text{fluffy}}, \underbrace{0}_{\text{gray}}]$$

2-D projection of output $\longrightarrow$

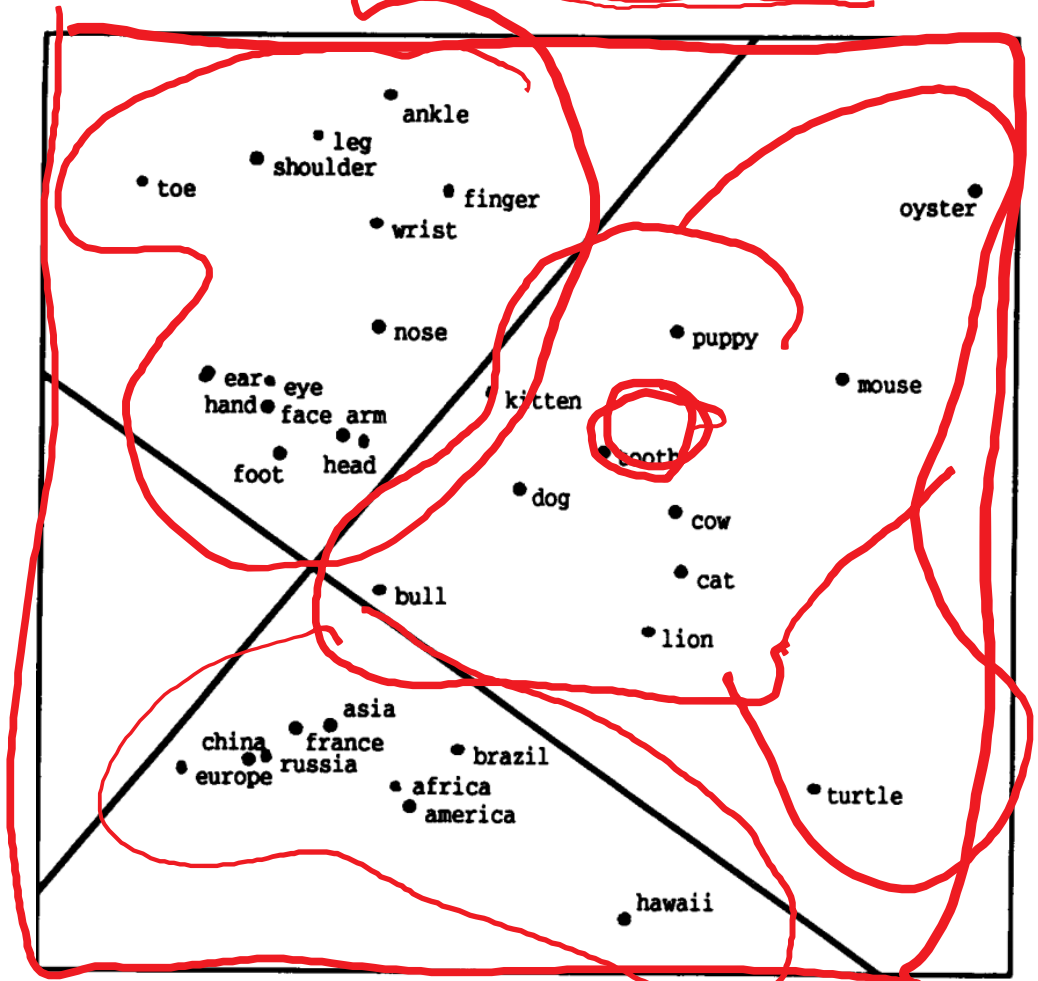
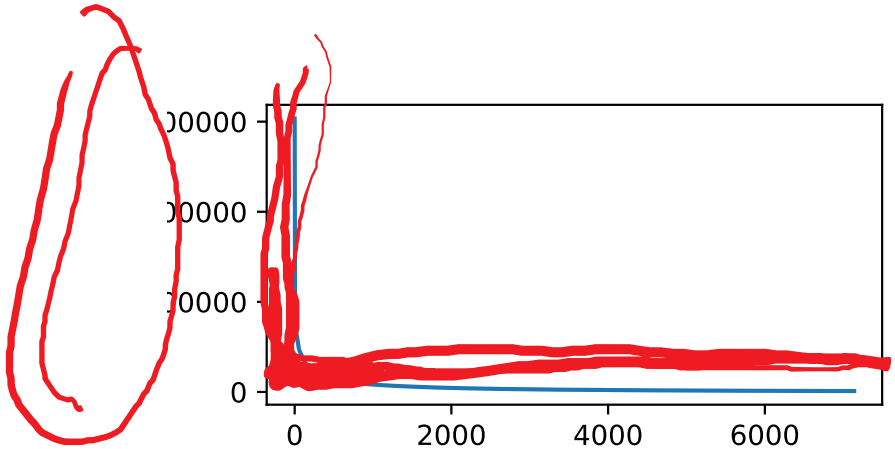


Figure 2. Multidimensional scaling of co-occurrence vectors.

Linguistics: Very few common words, a lot of rare words



Disclaimer: This histogram is after pre-processing, which removes common and rare words–real histogram is more pronounced

Pairwise mutual information (PPMI) matrix

$$\text{PMI}_{i,j} = \log\left(\frac{\text{corpus length} \times \#\text{ co-occurance of word } i \text{ with word } j}{\text{freq. of word } i \times \text{freq. of word } j}\right)$$

Positive pairwise mutual information (PPMI) matrix

$$\text{PPMI}_{i,j} = \max\{\text{PMI}_{i,j}, 0\}$$

Word embeddings

Other notable methods

- Deerwester, et al '90 (Latent semantic analysis)
- Bengio et al 03, Collobert and Weston 08 (Deep neural nets)
- Pennington, Socher, Manning '14 (GloVe), Mikolov, Chen, Corrado, Dean '13 (word2vec)

Levy and Goldberg '14: many modern methods can be interpreted as implicit matrix factorization of a specific matrix: $C = UU^T$

- $C \in \mathbb{R}^{n \times n}$ encodes similarity between all pairs of n words (n is large)
- $U \in \mathbb{R}^{n \times d}$, each row contains d-dim. word vector representation ($d \ll n$)

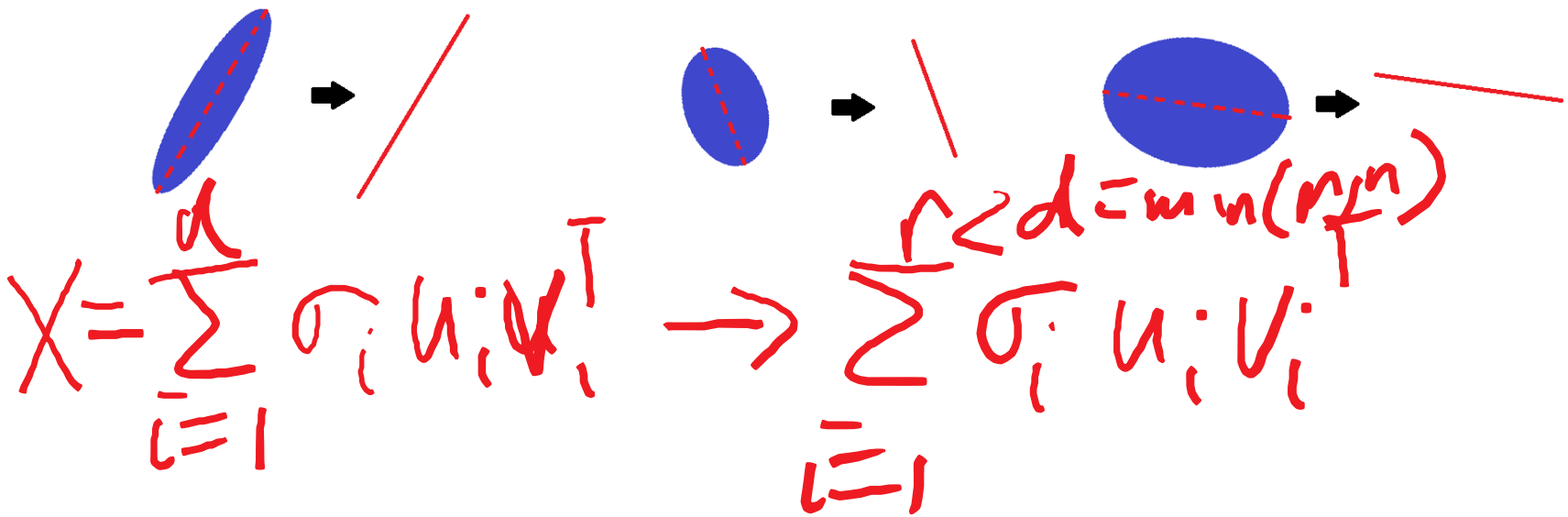
Principle component analysis

Principle component analysis

Given $X \in \mathbb{R}^{m \times n}$, find $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{n \times r}$ where

$$UV^T = \min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}} \|UV^T - X\|_F^2$$

Example: compress 2D to 1D



Eckart-Young-Mirsky theorem

- Given some $X \in \mathbb{R}^{m,n}$
- Take the singular value decomposition of X

$$X = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T,$$

where

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}}$$

are the singular values and correspond to singular vectors u_i, v_i.

- Then $\hat{X} := \sum_{i=1}^{r} \sigma_i u_i v_i^T$ is the best rank-r approximation of X; that is,

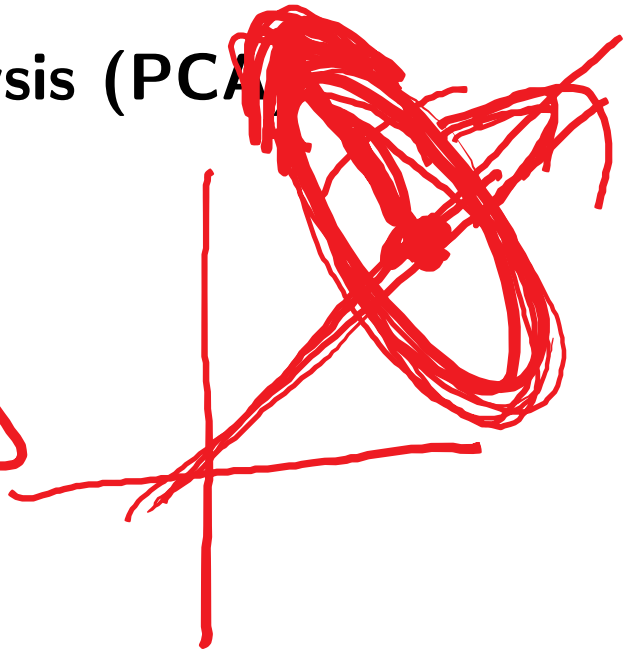
$$\|\hat{X} - X\|_\star = \min_{X' \text{ rank } r} \|X' - X\|_\star^2$$

where $\|\cdot\|_\star$ is the Frobenius or spectral norm

Principle component analysis (PCA)

- Given m centered data points $x_1, ..., x_m \in \mathbb{R}^n$

$$\sum_{i=1}^{n} x_i = 0$$



- Perform a SVD on the composed data matrix

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T, \quad \sigma_1 \geq \sigma_2 \geq \cdots$$

$C = \hat{U}\hat{U}^T$

- Reconstruct with some $r \ll n$

$$\hat{X} = \begin{bmatrix} \hat{x}_1^T \\ \vdots \\ \hat{x}_m^T \end{bmatrix} = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Reconstruction error

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i^T\|_2^2 = \|X - \hat{X}\|_F^2$$

$$= \| \sum_{i=r+1}^{\min\{m,n\}} \sigma_i u_i v_i^T \|_F^2$$

$$= \sum_{i=r+1}^{\min\{m,n\}} u_i^T \underbrace{XX^T}_{\text{gram matrix, } m \times m} u_i$$

$$= \sum_{i=r+1}^{\min\{m,n\}} v_i^T \underbrace{X^T X}_{\text{covariance matrix, } n \times n} v_i$$

[Turk, Pentland '91] Input images

[Turk, Pentland '91] Principle components

[Turk, Pentland '91] Reconstruction

Each image corresponds to adding 8 principle components

$$u = \hat{u} Q^T$$

Independent component analysis

recommendation → $R \approx U V^T$, $Q^T Q = I$

word embeddings → $C \approx U \Sigma U^T$

spectra → $X \approx U_r \Sigma_r V_r^T$

Blind source separation (BSS) problem

- d speakers, d sensors
- $x(t) = As(t)$, $A \in \mathbb{R}^{d \times d}$ is mixing matrix
- Given $x(t)$, can we recover $s(t)$?

Aside: chain rule for PDF of linear transformations

- Suppose that $s = Wx$ and W is invertible.
- If $p_X(x)$ is the probability distribution (pdf) of x, what is the pdf of s?

Aside: chain rule for PDF of linear transformations

- Suppose that $s = Wx$ and W is invertible.
- If $p_X(x)$ is the probability distribution (pdf) of x, what is the pdf of s?
  Ans:

$$p_S(s) = p_S(Wx) \underbrace{|W|}_{\text{determinant}}$$

- Why?

Aside: chain rule for PDF of linear transformations

- Suppose that $s = Wx$ and W is invertible.
- If $p_X(x)$ is the probability distribution (pdf) of x, what is the pdf of s?
  Ans:

$$p_S(s) = p_S(Wx) \underbrace{|W|}_{\text{determinant}}$$

- Why? Chain rule

1-D example: CDFs

$$F_S(s) = \Pr(S < s) = \Pr(wX < wx) = F_X(x)$$

Now differentiate

$$p_S(s) = \frac{d}{ds} F_S(s) = \frac{d}{ds} F_X(x) \overset{x=ws}{=} w p_X(x)$$

Formal ICA setup

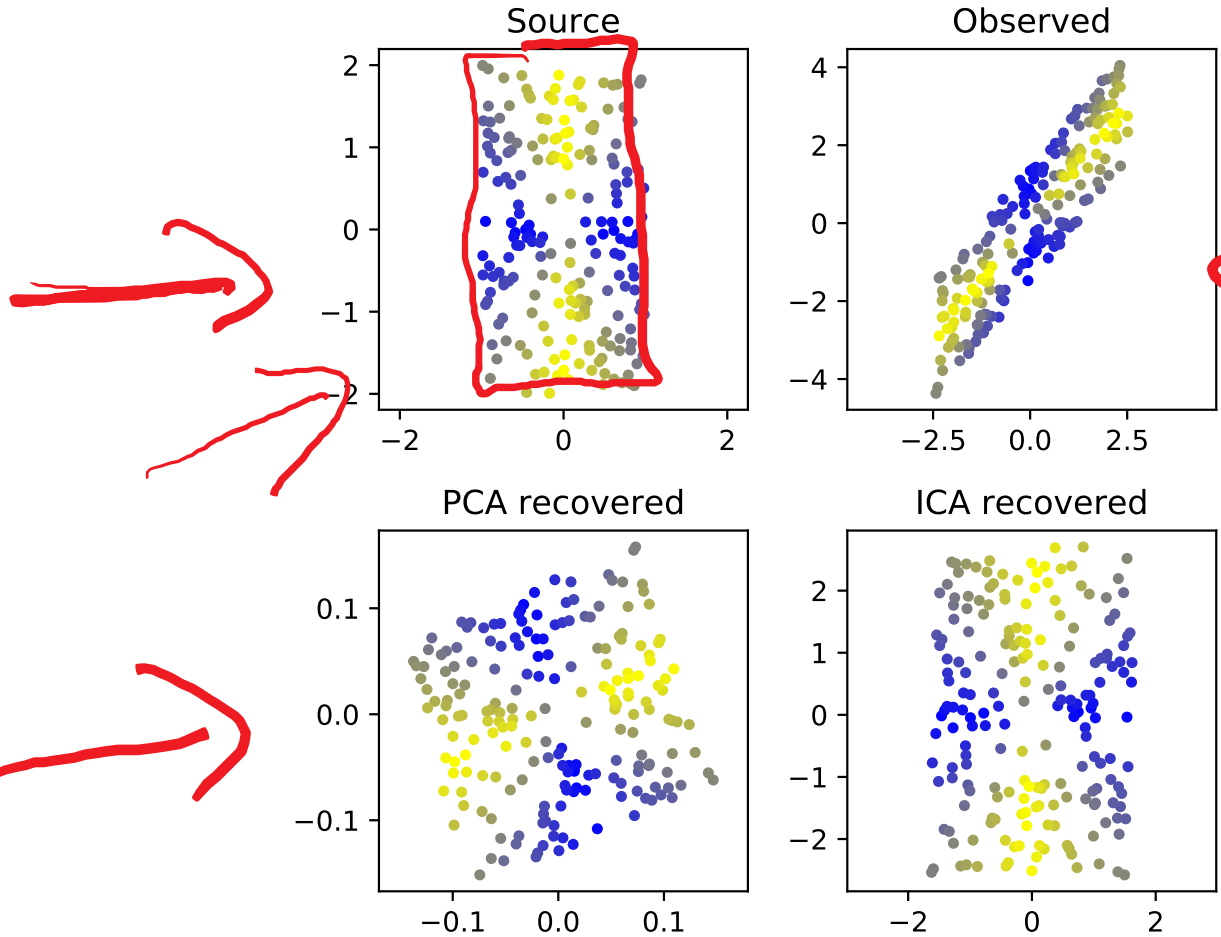
- Given $x_1, ..., x_T \in \mathbb{R}^d$, find $A \in \mathbb{R}^{d \times d}$ such that
  - $x_i = As_i$ for all i
  - $s_1, ..., s_T$ are all independent
- The MLE is

$$\frac{1}{T}\log(\Pr(W|x)) = \frac{1}{T}\underbrace{\sum_{i=1}^{T}}_{\text{independet observations}}\log(\Pr(W|x_i))$$

$$= \frac{1}{m}\sum_{i=1}^{T}\underbrace{\sum_{j=1}^{d}}_{\text{independent sources}}\log(p_S(w_j^T x_i)) + \underbrace{\log(|W|)}_{\text{chain rule}}$$

- Procedure: find W_{ICA} the maximizer of this loss function
- Find $(s_{\text{ICA}})_i = W_{\text{ICA}}^{-1} x_i$, $i = 1, ..., T$

Example: Uniform 2-D distribution



- PCA recovers linear subspace, but rotationally ambiguous
- ICA can recover subspace and rotation

ICA vs PCA on faces

(a) Basis Images corresponding to ICA Architecture I

(b) Basis Images corresponding to ICA Architecture II

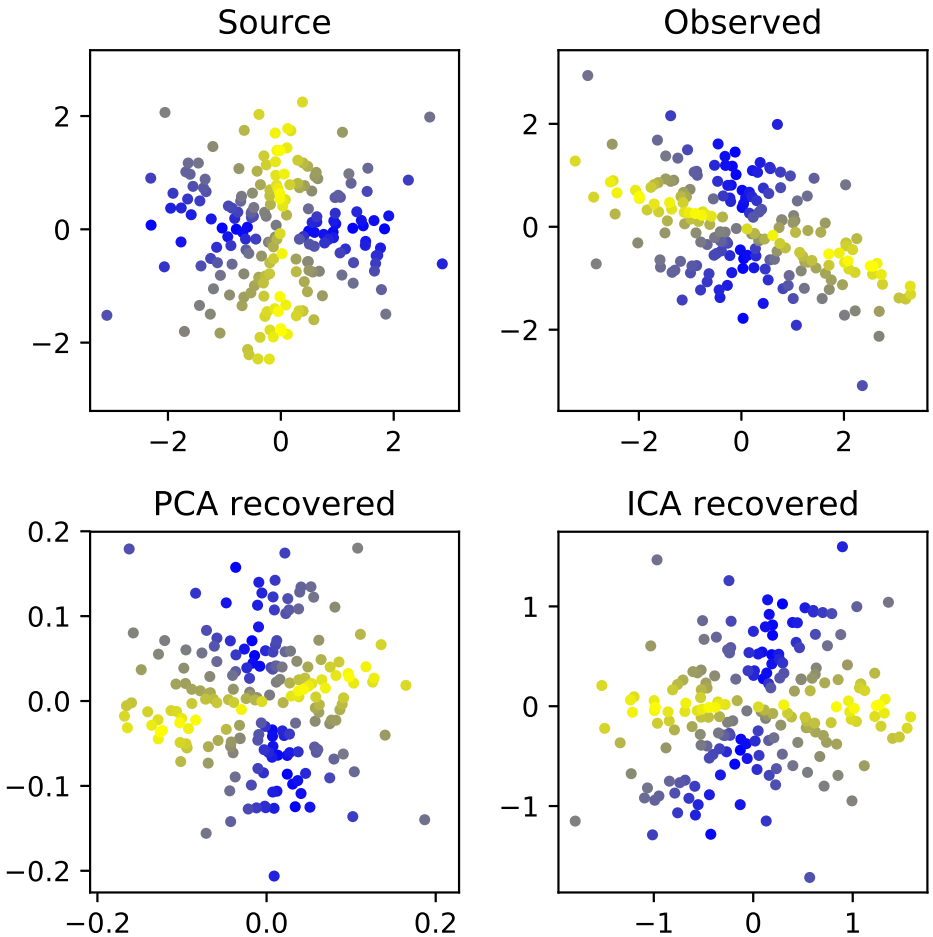
(c) Basis Images (eigenfaces) corresponding to PCA

Figure 2. Basis images corresponding to ICA Architecture I, ICA Architecture II and PCA

Top: ICA, Middle: ICA with whiten preprocessing, Bottom: PCA

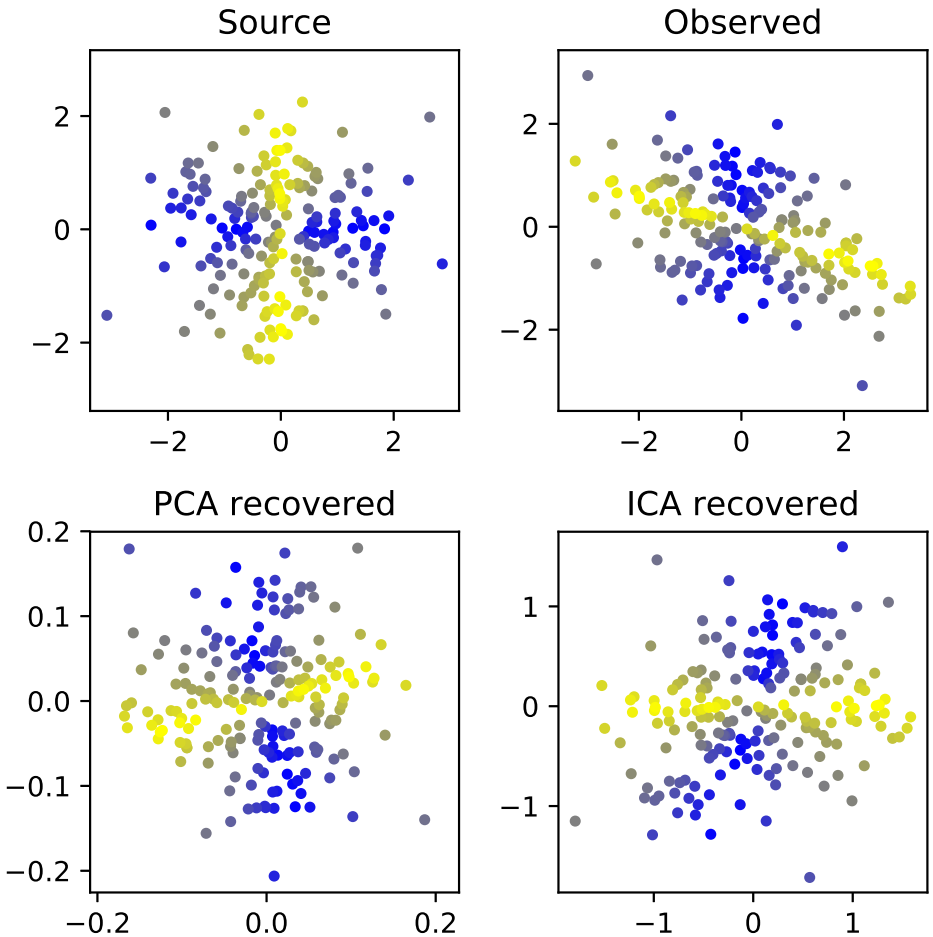
Yang, Zhang, Yang "Is ICA significantly better than PCA for face recognition?" ICCV 05

ICA cannot do better than PCA on Gaussian data



Why? (Just answer based on what you see in the picture)

ICA cannot do better than PCA on Gaussian data



Why? (Just answer based on what you see in the picture)

Gaussian i.i.d. = rotationally invariant. (Only distribution with this property.)