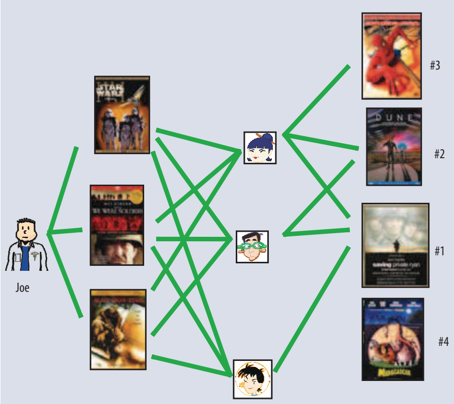
Lesson, Latent Factors and SVD

Analytics Lab, BIPM

Recommendation Engines

- **content filtering** approach
  - creates a profile for each user or product (a movie profile could include attributes regarding its genre, the participating actors, its box office popularity, and so forth. User profiles might include demographic information or answers provided on a suitable questionnaire.)
  - associate users with matching products.
  - requires gathering external information that might not be available or easy to collect.
  - Music Genome Project, Pandora.com
- **collaborative filtering** relies only on past user behavior
  - for example, previous transactions or product ratings
  - not requiring the creation of explicit profiles.
  - "cold start problem": inability to address new products and users.

Collaborative filtering: neighborhood methods



The user-oriented neighborhood method. Joe likes the three movies on the left. To make a prediction for him, the system finds similar users who also liked those movies, and then determines which other movies they liked. In this case, all three liked Saving Private Ryan, so that is the first recommendation. Two of them liked Dune, so that is next, and so on.

Collaborative filtering: latent factor models

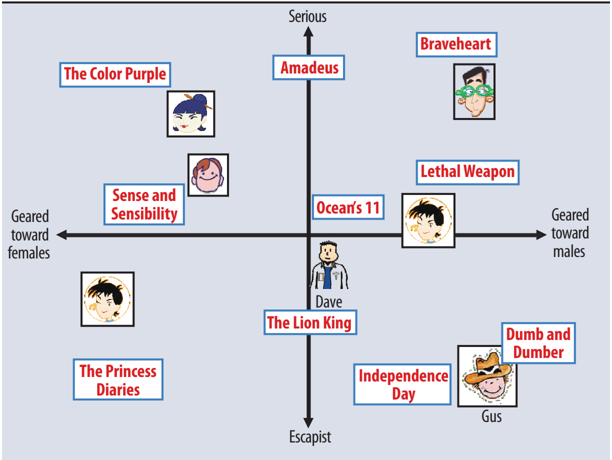
- Tries to explain the ratings by characterizing both items and users on “few” (e.g. 20 to 100) factors inferred from the ratings patterns.
- Such **latent factors** alternative to e.g. human created song genres.
- For movies, the discovered factors might measure
  - obvious dimensions such as comedy versus drama, amount of action, or orientation to children;
  - less well-defined dimensions such as depth of character development or quirkiness;
  - or completely uninterpretable dimensions.
- For users, each factor measures how much the user likes movies that score high on the corresponding movie factor.

The Netflix competition I

Netflix provided 100M ratings (from 1 to 5) of 17K movies by 500K users. Task: for (User,Movie,?) not in the database, predict what the Rating would be–that is, predict how the given User would rate the given Movie.

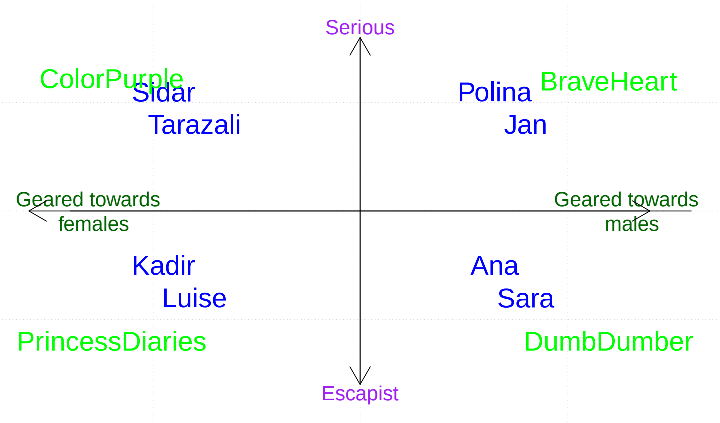
Imagine for a moment that we have all 8.5 billion (!) ratings (and a lot of weary users). Presumably there are some generalities to be found in there, something more concise and descriptive than 8.5 billion completely independent and unrelated ratings. For instance, any given movie can, to a rough degree of approximation, be described in terms of some basic attributes such as overall quality, whether it's an action movie or a comedy, what stars are in it, and so on. And every user's preferences can likewise be roughly described in terms of whether they tend to rate high or low, whether they prefer action movies or comedies, what stars they like, and so on. And **if those basic assumptions are true, then a lot of the 8.5 billion ratings ought to be explainable by a lot less than 8.5 billion numbers**, since, for instance, a single number specifying how much action a particular movie has may help explain why a few million action-buffs like that movie.

Latent factor models



A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes-male versus female and serious versus escapist.

No hidden (latent) factors



Only Movie Rankings available

	PrincessDiaries	DumbDumber	ColorPurple	BraveHeart
Luise	4	2	2	1
Sara	2	4	1	2
Tarazali	2	1	4	2
Jan	1	2	2	4
Kadir	4	2	3	1
Ana	2	4	1	3
Sidar	2	0	5	3
Polina	0	2	3	5

The Netflix competition II

We'll assume that a user's rating of a movie is composed of a sum of preferences about the various aspects of that movie.

For example, imagine that we limit it to forty aspects, such that each movie is described only by forty values saying how much that movie exemplifies each aspect, and correspondingly each user is described by forty values saying how much they prefer each aspect. To combine these all together into a rating, we just multiply each user preference by the corresponding movie aspect, and then add those forty leanings up into a final opinion of how much that user likes that movie. E.g., Terminator might be (action=1.2,chickflick=-1,. . .), and user Joe might be (action=3,chickflick=-1,. . .), and when you combine the two you get Joe likes Terminator with \$3*1.2 + -1*-1 + . . . = 4.6+. . . \$. Note here that Terminator is tagged as an anti-chickflick, and Joe likewise as someone with an aversion to chickflicks, so Terminator actively scores positive points with Joe for being decidedly un-chickflicky. (Point being: negative numbers are ok.) Anyway, all told that model requires $40 * (17K + 500K)$ values, or about $20M$ – 400 times less than the original $8.5B$.

Matrix Factorization

Matrix factorization models map both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Accordingly, each item i is associated with a vector $q_i \in R^f$, and each user u is associated with a vector $p_u \in R^f$ such that the dot product approximates the user u's rating of item i:

$$\hat{r}_{ui} = q_i^T \cdot p_u$$

Singular Value Decomposition (SVD)

$$X = U \cdot S \cdot V^T$$

$$\underset{m \times n}{\overset{\hat{X}}{\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} \approx \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \dots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \dots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

Figure 1: SVD with r=min(n,m) (http://tinyurl.com/yyrpbbcq)

SVD has the minimal reconstruction Sum of Square Error (SSE)

$$SSE_k = \sum_{i,j \in X} \left(X_{ij} - [U_{m \times k} \cdot S_{k \times k} \cdot V^T_{n \times k}]_{ij} \right)^2$$

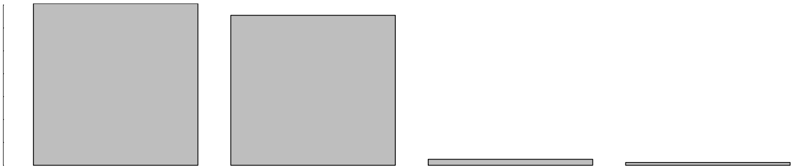
Left Singular Vectors

	U1	U2	U3	U4
Luise	-1.45	1.10	-0.36	0.13
Sara	-1.45	-1.10	0.36	0.13
Tarazali	0.72	1.07	0.36	0.21
Jan	0.72	-1.07	-0.36	0.21
Kadir	-1.09	1.45	0.00	-0.22
Ana	-1.09	-1.45	-0.00	-0.22
Sidar	1.81	1.45	0.00	-0.11
Polina	1.81	-1.45	-0.00	-0.11

Right Singular Vectors

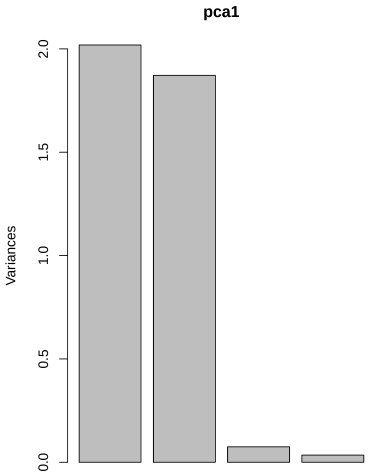
	V1	V2	V3	V4
PrincessDiaries	-0.49	0.51	-0.49	-0.51
DumbDumber	-0.49	-0.51	0.49	-0.51
ColorPurple	0.51	0.49	0.51	-0.49
BraveHeart	0.51	-0.49	-0.51	-0.49

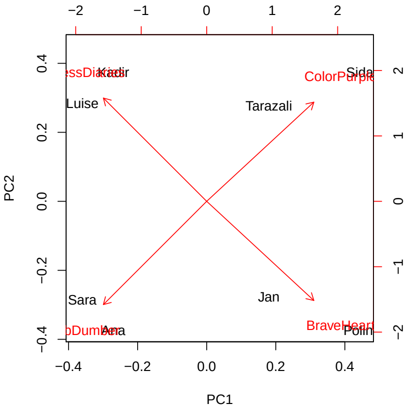
Singular Values



- Missing Values: Stochastic gradient descent
- Regularization

PCA and latent factors



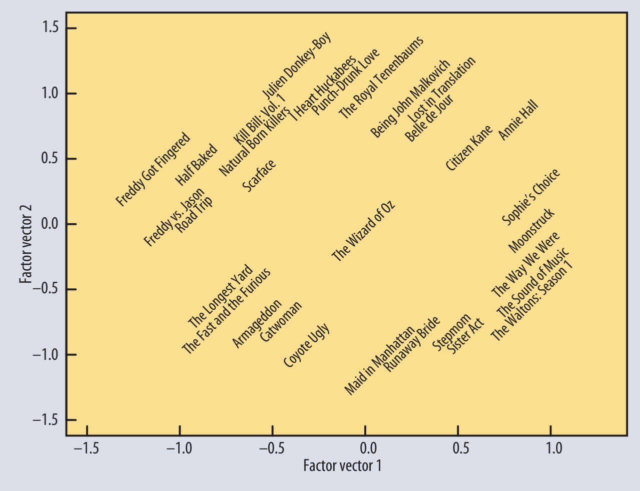


PCA and SVD

loadings and projections

	PC1	PC2
PrincessDiaries	-0.49	0.51
DumbDumber	-0.49	-0.51
ColorPurple	0.51	0.49
BraveHeart	0.51	-0.49
Luise	-1.45	1.10
Sara	-1.45	-1.10
Tarazali	0.72	1.07
Jan	0.72	-1.07
Kadir	-1.09	1.45
Ana	-1.09	-1.45
Sidar	1.81	1.45
Polina	1.81	-1.45

Netflix prize



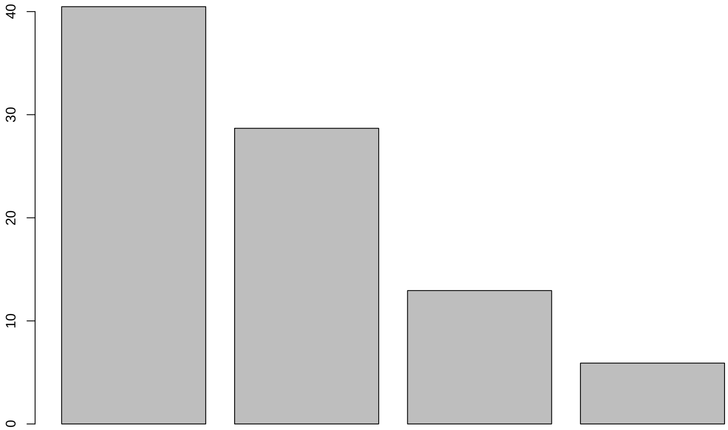
The first two vectors from a matrix decomposition of the Netflix Prize data. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

Interpretation

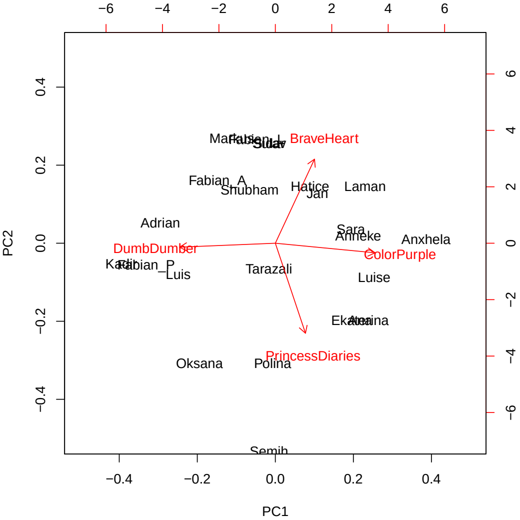
Someone familiar with the movies shown can see clear meaning in the latent factors. The first factor vector (x-axis) has on one side lowbrow comedies and horror movies, aimed at a male or adolescent audience (*Half Baked*, *Freddy vs. Jason*), while the other side contains drama or comedy with serious undertones and strong female leads (*Sophie's Choice, Moonstruck*). The second factorization axis (y-axis) has independent, critically acclaimed, quirky films (*Punch-Drunk Love, I Heart Huckabees*) on the top, and on the bottom, mainstream formulaic films (*Armageddon, Runaway Bride*). There are interesting intersections between these boundaries: On the top left corner, where indie meets lowbrow, are *Kill Bill* and *Natural Born Killers*, both are movies that play off violent themes. On the bottom right, where the serious female-driven movies meet the mainstream crowd-pleasers, is *The Sound of Music*. And smack in the middle, appealing to all types, is *The Wizard of Oz*. In this plot, some movies neighboring one another typically would not be put together. For example, *Annie Hall* and *Citizen Kane* are next to each other. Although they are stylistically very different, they have a lot in common as highly regarded classic movies by famous directors. Indeed, the third dimension in the factorization does end up separating these two.

BIPM movie ranking

```
## [1] "d" "u" "v"
```



Our own latent preferences



Latent Semantic Analyis (LSA)

https://technowiki.wordpress.com/2011/08/27/latent-semantic-analysis-lsa-tutorial