PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA is probably the most common "dimension reduction" tool in data analysis. As an example we use the Netflix Challenge, a dataset of ratings, on a 1–5 scale, of about 18000 films by 500000 people. Here's how this data table might look:

	Film 1 (Ace Ventura)	2 (The Alamo)	3 (Avatar) ...
person 1 (Alice)	4	2	5 ...
person 2 (Bob)	1	5	4
⋮	⋮	⋮	⋮ ⋱

$M = 500000$ rows $N = 18000$ columns

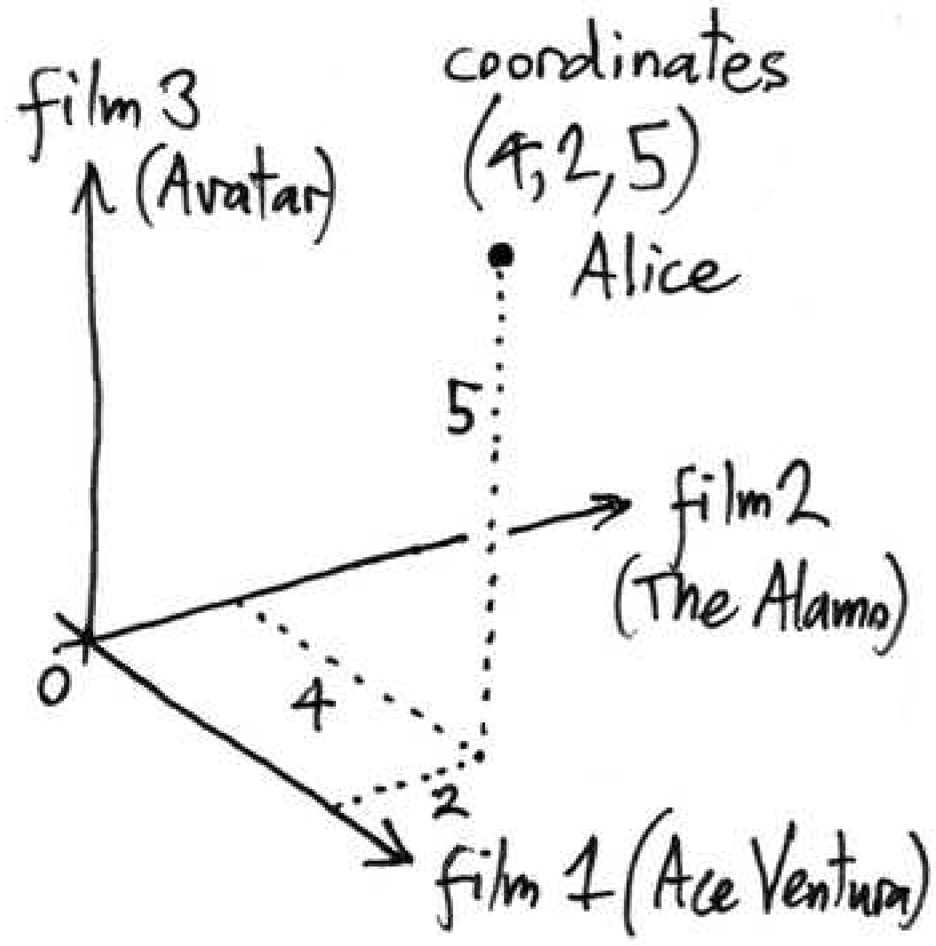
This is a <u>huge</u> matrix, call it A, with entries a_{ij} person index, film index.

To explain PCA, imagine everyone rated every movie, so that A is entirely known. The idea behind PCA is that Alice's high rating for Avatar ($a_{1,3} = 5$) is controlled by an unknown number of "latent" factors that both shape Alice's taste (eg, she likes sci-fi, dislikes violence) <u>and</u> describe films (Avatar is futuristic but nonviolent). Bob's taste differs (he likes sci-fi, but moreso violence), "explaining" his higher rating for The Alamo over Avatar.

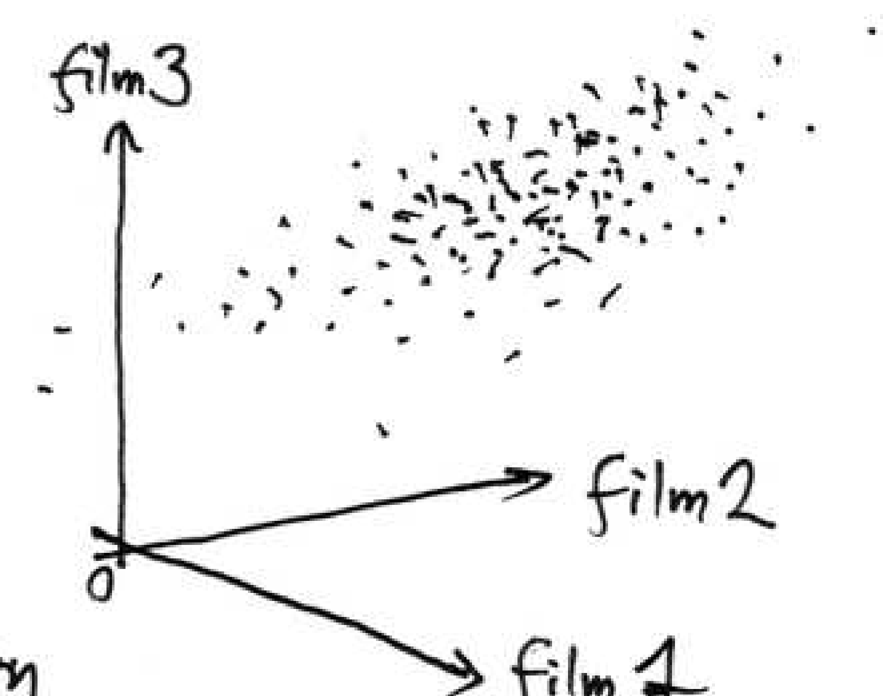
PCA extracts these factors ("futuristic", "violent", etc), ranking them most to least important, by analyzing the matrix A.

Let's plot all Alice's ratings as a single point in 3D "ratings space":

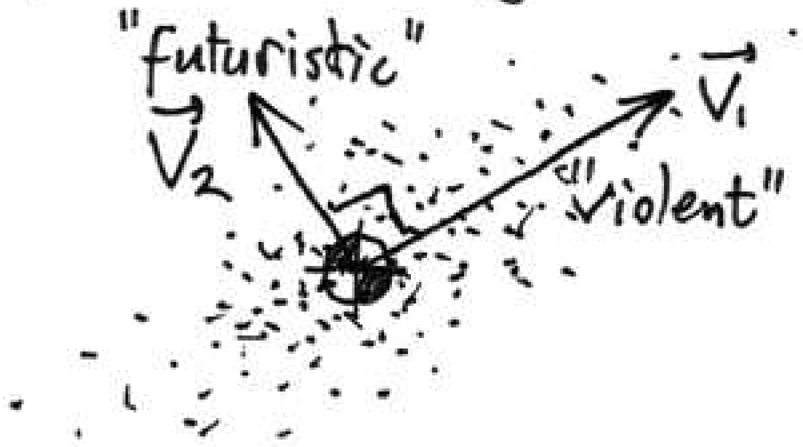
> In fact there's 18000 dimensions, but we can only sketch the first 3!



Now let's add everyone else:
Each row of the table is a point.
This cloud of 500000 points is equivalent to the matrix A.



PCA extracts the crude <u>geometry</u> of this point cloud: the 1st "principal component" (eigenvector $\vec{v}_1$) is the cloud's longest <u>axis</u>, ie the factor explaining the most <u>variance</u> in ratings. The 2nd P.C. is the direction, $\vec{v}_2$, at right angles to $\vec{v}_1$, of most <u>remaining</u> variance, and so on. The hope is that the gross shape of the cloud is captured by a <u>few</u> directions of spread, even though it lives in a <u>huge</u> dimension space.



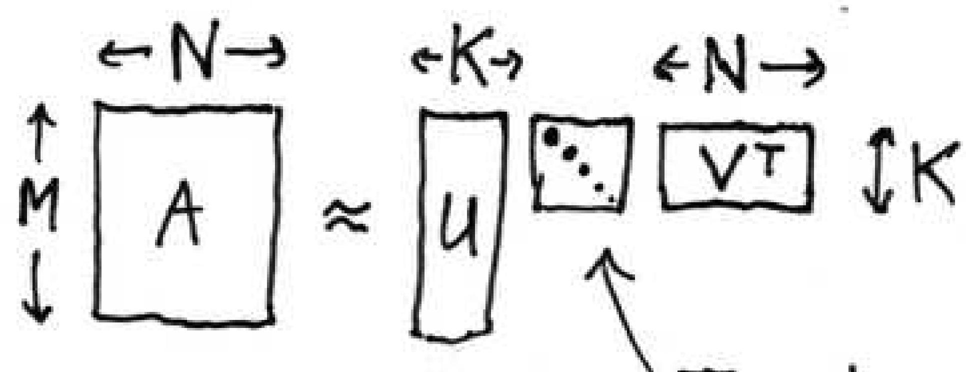
- A note on mean subtraction: the P.C. vectors $\vec{v}_1, \vec{v}_2$ are sketched emanating from the cloud's "center of mass" ⊕. This is because Avatar, for example, may have a higher average rating than other films; this is not a "latent" effect. To remove these film-specific effects, each column of the matrix A has its average subtracted before doing PCA. Geometrically, this shifts the origin from "0" to ⊕, centering the cloud. Likewise, since Alice may be universally more generous than Bob, for example, row means are usually also subtracted.

With that intuitive picture complete, here are the formulae! PCA performs a (partial) "singular value decomposition" (SVD) of A, writing it as the product of 3 matrices:

$$A \approx U \Sigma V^T$$

$$V = \begin{bmatrix} \downarrow & \downarrow & \cdots \\ \vec{v}_1 & \vec{v}_2 & \end{bmatrix}$$

stack of eigenvectors

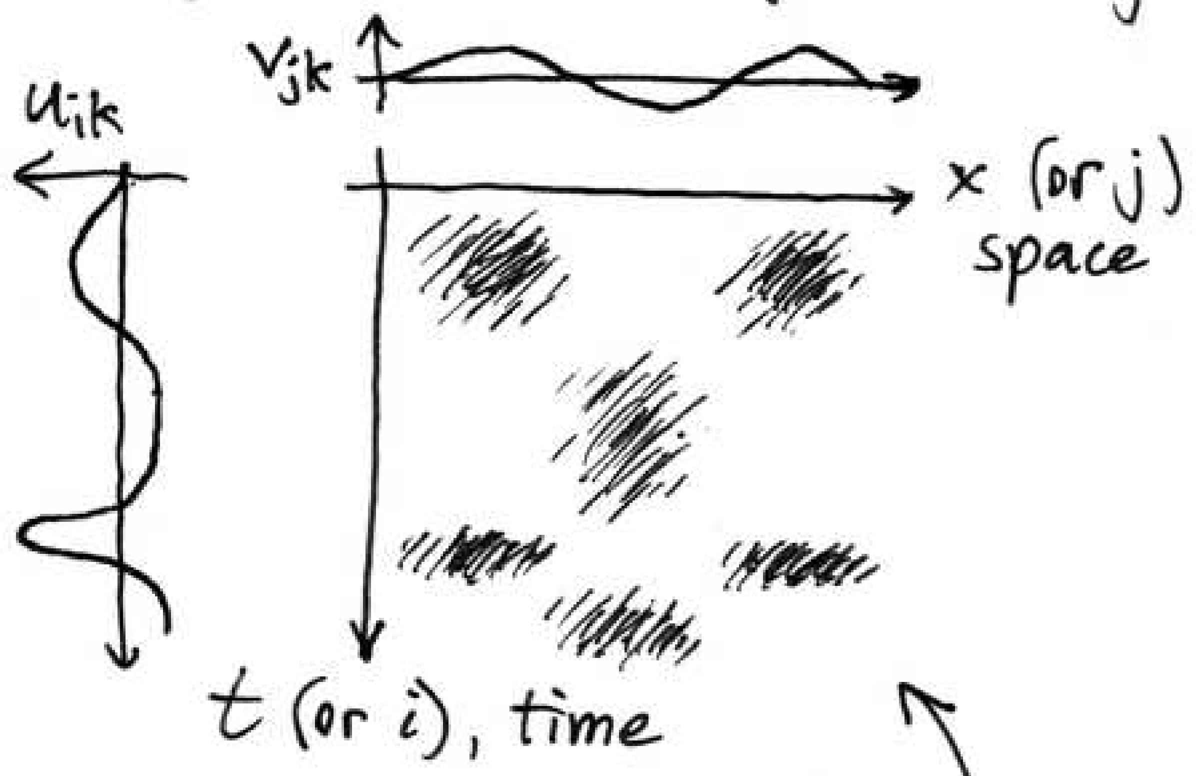


Σ, diagonal matrix with entries $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_K$ giving importance of each factor.

K = number of factors

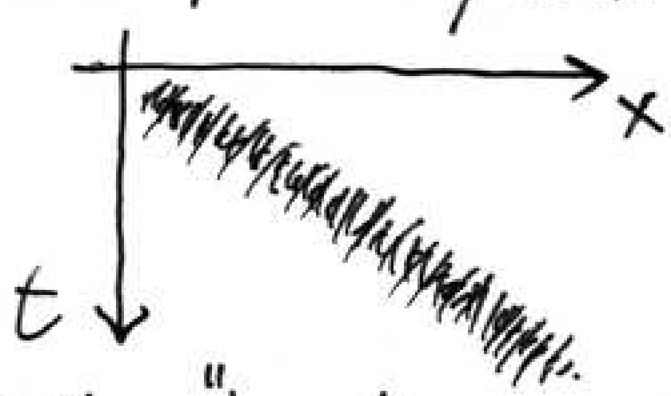
Usually K is small (less than a few dozen). PCA builds the best rank-K approximation to the data A.

- Connection to "empirical orthogonal functions" (EOF): Replace the film (column) label j by space, and the person (row) label i by time. PCA then extracts, from data such as temperature recorded over space and time, the dominant temperature "modes". This is very common in geophysical & climate analysis, and called EOF. Writing the SVD as $a_{ij} \approx \sum_{k=1}^{K} \sigma_k u_{ik} v_{jk}$, each term is a <u>separable</u> mode, ie a product of a function of space only (v_{jk}), and a function of time only (u_{ik}), like this:



Note "checkerboard" pattern.

This is a possible mode in EOF. For contrast, here's a <u>non</u>separable function:



It's a "traveling wave", which EOF struggles with!

- The Netflix saga:

In fact, only 1% of the entries a_{ij} were known (99% of films were unrated by the average person). This makes the task (a low-rank "matrix completion" problem) a Challenge, more than plain PCA. The "training set" of known entries was still huge (100 million entries). The \$1M prize was given in 2009 for the algorithm to first reduce, by 10%, the prediction error on a hidden "test set" of 3 million entries. Latent factor (PCA-based) models played a huge role in successful algorithms, and continue to do so in "collaborative filtering" (online recommendation systems).