

MUSIC RECOMMENDATION AT SPOTIFY

- Manual Curation



- Manually Tag Attributes



- Audio Content,
Metadata, Text Analysis



- Collaborative Filtering



EXAMPLE OF CONTENT-BASED RECOMMENDATION

The “Music Genome Project” of Pandora

- ▶ Up to 450 “genes”
- ▶ Each gene corresponds to a characteristic of the music
- ▶ Human-powered: Each song is analyzed by a musicologist



Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=22s>

Alice = 95% Romance fan + 5% Action fan + ...

Bob = 90% Romance fan + 10% Action fan + ...

Carol = 5% Romance fan + 95% Action fan + ...

...

Problem motivation

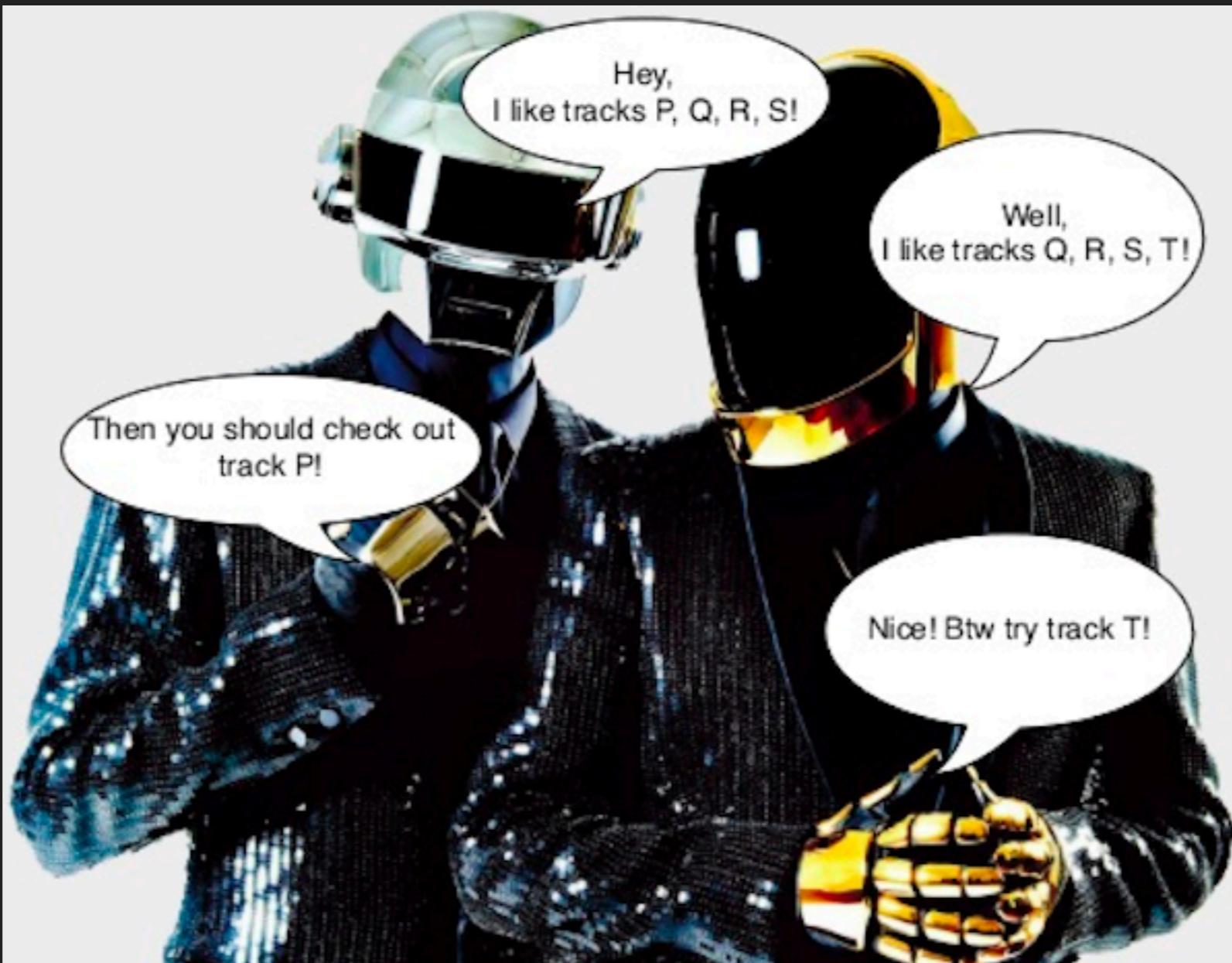
Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=22s>

Optimization algorithm:

$$\min_{\theta^{(1)}, \dots, \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^n (\theta_k^{(j)})^2$$

COLLABORATIVE FILTERING



AGGREGATING THE NEIGHBORS' RATINGS

?	2	?	2	5	?	4	Alice
1	1	?	3	5	?	?	Bob
			:				:
4	5	?	?	1	4	2	Zoe

Alice is close to Bob. Her rating for Titanic is probably close to Bob's (1). So $?$ $\leftarrow 1$

COLLABORATIVE FILTERING: NEIGHBORHOOD METHOD

How to define similarity?

- ▶ Number of common related items
- ▶ Average absolute difference between ratings
- ▶ Cosine angle between vectors u and v
- ▶ Pearson correlation coefficient
- ▶ And: millions of others!

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0
Romance forever	5	?	?	0	1.0	0.01
Cute puppies of love	?	4	0	?	0.99	0
Nonstop car chases	0	0	5	4	0.1	1.0
Swords vs. karate	0	0	5	?	0	0.9

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=22s>

Alice = 95% Romance fan + 5% Action fan + ...

Bob = 90% Romance fan + 10% Action fan + ...

Carol = 5% Romance fan + 95% Action fan + ...

...

COLLABORATIVE FILTERING

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (romance)	x_2 (action)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=22s>

Alice = ? % Romance fan + ? % Action fan + ...

Bob = ? % Romance fan + ? % Action fan + ...

Carol = ? % Romance fan + ? % Action fan + ...

...

COLLABORATIVE FILTERING

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (re?)	x_2 (re?)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=22s>

Alice = ? % | ? fan + ? % | ? fan + ...

Bob = ? % | ? fan + ? % | ? fan + ...

Carol = ? % | ? fan + ? % | ? fan + ...

...

MATRIX FACTORIZATION

- A lot of hype during the Netflix Prize
(2006-2009: *improve our system, get rich*)
- Model the ratings in an insightful way
- Takes its root in dimensional reduction and **SVD**

Slides by Nicholas Hug

BEFORE SVD: PCA

- Here are 400 greyscale images (64 x 64):



- Put them in a 400×4096 matrix X :

$$X = \begin{pmatrix} & \text{Face 1} & \\ & \text{Face 2} & \\ & \vdots & \\ & \text{Face 400} & \end{pmatrix}$$

Slides by Nicholas Hug

BEFORE SVD: PCA

PCA will reveal 400 of those creepy **typical** guys:



These guys can build back all of the original faces

$$\begin{aligned}\text{Face } 1 &= \alpha_1 \cdot \text{Creepy guy } \#1 \\ &\quad + \alpha_2 \cdot \text{Creepy guy } \#2 \\ &\quad + \dots \\ &\quad + \alpha_{400} \cdot \text{Creepy guy } \#400\end{aligned}$$

PCA also gives you the α_i .

In advance: you don't need all the 400 guys



Slides by Nicholas Hug

PCA ON A RATING MATRIX? SURE!

Assume all ratings are **known**

$$X = \begin{pmatrix} & \text{Face 1} & \\ & \text{Face 2} & \\ & \vdots & \\ & \text{Face 400} & \end{pmatrix} \quad R = \begin{pmatrix} & \text{Alice} & \\ & \text{Bob} & \\ & \vdots & \\ & \text{Zoe} & \end{pmatrix}$$

Exact same thing! We just have ratings instead of pixels.

PCA will reveal **typical users**.



Alice = 10% **Action fan** + 10% **Comedy fan** + 50% **Romance fan** + ...

Bob = 50% **Action fan** + 30% **Comedy fan** + 10% **Romance fan** + ...

Zoe = ...

Slides by Nicholas Hug

COLLABORATIVE FILTERING: MATRIX FACTORIZATION

PCA ON A RATING MATRIX? SURE!

Assume all ratings are **known**. Transpose the matrix

$$X = \begin{pmatrix} & \text{Face 1} & \\ & \text{Face 2} & \\ \vdots & & \\ & \text{Face 400} & \end{pmatrix} \quad R^T = \begin{pmatrix} & \text{Titanic} & \\ & \text{Toy Story} & \\ \vdots & & \\ & \text{Fargo} & \end{pmatrix}$$

Exact same thing! PCA will reveal **typical movies**.



Titanic = 20% **Action** + 0% **Comedy** + 70% **Romance** + ⋯

Toy Story = 30% **Action** + 60% **Comedy** + 0% **Romance** + ⋯

Note: in practice, the factors semantic is not clearly defined.

Slides by Nicholas Hug

SVD IS PCA²

- PCA on R gives you the typical **users** U
- PCA on R^T gives you the typical **movies** M
- SVD gives you **both** in one shot!

$$R = M\Sigma U^T$$

Σ is diagonal, it's just a scaler.

$$R = MU^T$$

This is our **matrix factorization!**

COLLABORATIVE FILTERING: MATRIX FACTORIZATION

Problem motivation

Movie	Alice (1)	Bob (2)	Carol (3)	Dave (4)	x_1 (re: ? :e)	x_2 (? :)
Love at last	5	5	0	0	?	?
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	?	?

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=22s>

$$\text{Alice} = ? \% \quad ? \quad \text{fan} + ? \% \quad ? \quad \text{fan} + \dots$$

$$\text{Bob} = ? \% \quad ? \quad \text{fan} + ? \% \quad ? \quad \text{fan} + \dots$$

$$\text{Carol} = ? \% \quad ? \quad \text{fan} + ? \% \quad ? \quad \text{fan} + \dots$$

...

Implicit Matrix Factorization

- Aggregate all (user, track) streams into a large matrix
- Goal:** Approximate binary preference matrix by inner product of 2 smaller matrices by minimizing the weighted RMSE (root mean squared error) using a function of plays, context, and recency as weight

$$\text{Users} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \approx \underbrace{\begin{pmatrix} x \\ \vdots \\ x \end{pmatrix}}_f \cdot \begin{pmatrix} \quad & Y & \quad \end{pmatrix} \} f$$

Songs

$$\min_{x,y} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i - \beta_u - \beta_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

- $p_{ui} = 1$ if user u streamed track i else 0
- $c_{ui} = 1 + \alpha r_{ui}$
- x_u = user u 's latent factor vector
- y_i = item i 's latent factor vector
- β_u = bias for user u
- β_i = bias for item i
- λ = regularization parameter

[1] Hu Y. & Koren Y. & Volinsky C. (2008) Collaborative Filtering for Implicit Feedback Datasets 8th IEEE Intern

Slides by Chris Johnson

COLLABORATIVE FILTERING

- ▶ Benefits from large user bases

The more users (ratings) the better!

- ▶ Flexible across different domains

Seemingly unrelated items by features

- ▶ Produces more serendipitous recommendations

Lovely surprises

- ▶ Capture more nuance around items

Implicitly capture features that cannot be easily quantified

COLLABORATIVE FILTERING

- ▶ Complexity and expense

“ Millions of items, and millions of users... ”

- ▶ Data sparsity

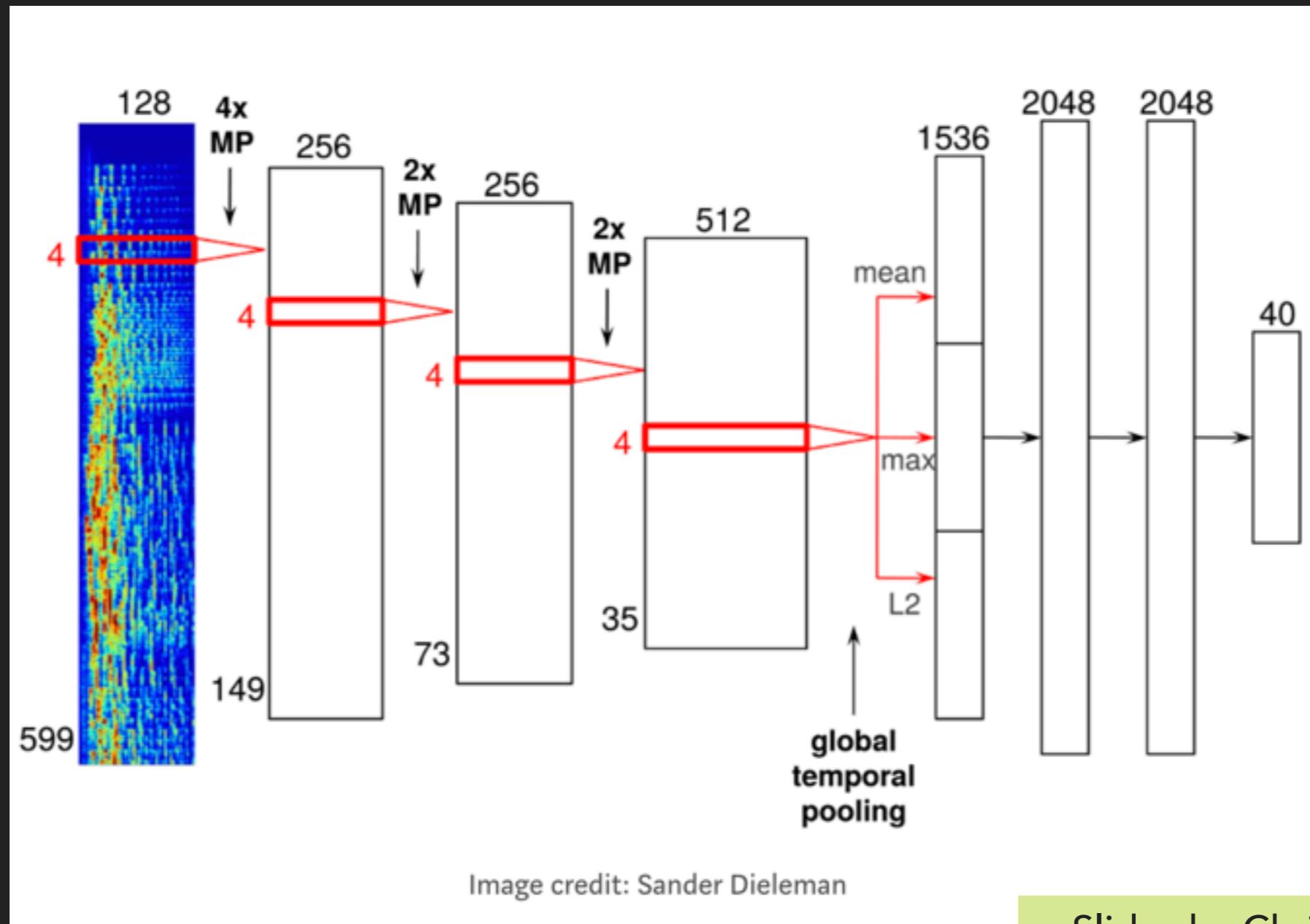
“ You have listened to only 0.0001% of what we have here ”

- ▶ The “cold-start” problem

“ You are too new for us to know you well :(”

SOLVING “COLD-START”

Raw audio: Convolutional Neural Networks



SOLVING “COLD-START”

Metadata & internet sources: Natural Language Processing

IN NATURAL LANGUAGE PROCESSING, THERE'S A TECHNIQUE CALLED **WORD2VEC**, WHICH TAKES WORDS AND ENCODES THEM INTO A MATHEMATICAL REPRESENTATION—A VECTOR. IN THESE MATHEMATICAL REPRESENTATIONS, VECTORS WITH A SIMILAR SHAPE WOULD EQUATE TO WORDS WITH A SIMILAR MEANING. BASICALLY, IT'S MATHEMATICAL REPRESENTATION OF THE IMPLICIT ASSOCIATIONS AND RELATIONSHIPS BETWEEN WORDS THAT WE KNOW TO BE TRUE IN EVERYDAY SPEECH.

WHAT SPOTIFY DOES IS VERY SIMILAR TO **WORD2VEC**. IT TAKES PLAYLISTS AND TREATS THEM AS A PARAGRAPH OR BIG BLOCK OF TEXT, AND TREATS EACH SONG IN THE PLAYLIST AS AN INDIVIDUAL WORD. THIS RESULTS IN VECTOR REPRESENTATIONS OF SONGS THAT CAN BE USED TO DETERMINE TWO PIECES OF MUSIC THAT ARE SIMILAR. AS SUCH, SPOTIFY IS ABLE TO DETERMINE WHICH SONGS ARE SIMILAR TO EACH OTHER, THUS ENABLING IT TO TACKLE THE COLD START PROBLEM AND RECOMMEND SONGS WITH VERY FEW PLAYS.

NLP Models on News and Blogs

The screenshot shows the Rolling Stone website. At the top, there are social sharing links for Like, Folio, and YouTube. The main navigation menu includes Music, Politics, TV, Movies, Culture, Sports, Reviews, Lists, RS Country, and Coverwall. A search bar is located at the top right. Below the menu, the title of the article is "Jack White vs. the Black Keys: A Beef History". A sub-headline reads: "As feud between rival blues-rock camps makes headlines again, we look back at story so far". The author's name is Andy Greene, and the date is September 14, 2015. Below the article, there are social sharing buttons for Facebook, Twitter, and Email. A large photo of the two artists is displayed.

Jack White vs. the Black Keys: A Beef History

As feud between rival blues-rock camps makes headlines again, we look back at story so far

BY ANDY GREENE, KORY GROW, BRITTANY SPANOS, PATRICK DOYLE,
HANK SHTEAMER September 14, 2015

Share Tweet Share Comment Email



RELATED

- 10 Craziest Beefs in MTV VMA History
- The 10 Wildest Rap Beefs of All Time
- 10 Craziest Musician Feuds

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

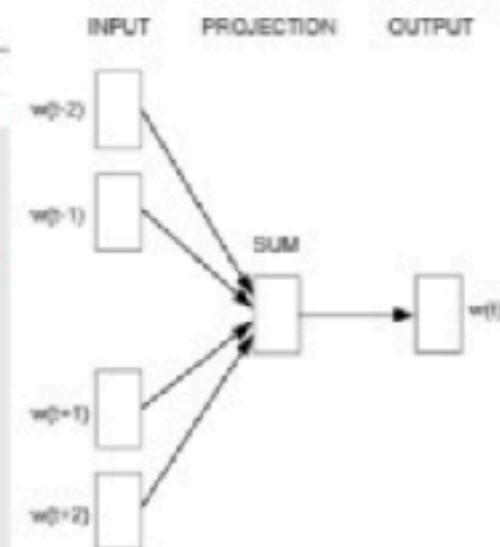
Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

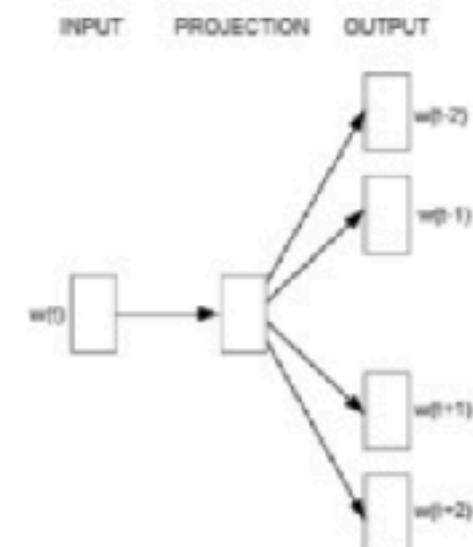
Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

The screenshot shows the Pitchfork website. The top navigation bar includes links for NEWS, THE PITCH, REVIEWS, TRACKS, FEATURES, PITCHFORKTV, PITCHRADIO, BEST IN MUSIC, STAFFLISTS, ARTISTS, FESTIVALS, and TIPS. Below the navigation, a featured review for "Ought - Sun Coming Down" is displayed. The review is by Matt Bernins and was published on September 15, 2015. It includes a photo of the band, a link to the album on Amazon MP3 & CD, and social sharing buttons for Twitter, Facebook, and Email. A sidebar on the right shows "PLAYING NEAR YOU" for Ought and "Most Read Album Reviews" for FOALS and SO DAYZ. A note at the bottom states: "My music was built on a whole lotta 'youth.' Because saying 'youth' is the most naked, innocuous form of rebellion—there is no easier way of showing you don't give a fuck than shirking the responsibility required to articulate the 'x' in 'you.' Seeing it through to the end is a show of diligence and commitment. So when Ought singer-guitarist Tim Darcy drops a clause 'youth' in the middle of 'Beautiful Blue Sky'—the spen-tacular centerpiece track of his band's sophomore album, Sun Coming Down—it's like he never left. Amid a song whose chorus reads like a laundry list of mid-century highlights ("Wings/Code/Now



CBOW



Skip-gram

Slides by Chris Johnson

NLP Models work great on Playlists!

Playlist itself is a document →

The screenshot shows a Spotify playlist page for 'SHOEGAZE'. The title 'SHOEGAZE' is at the top, followed by a subtitle: 'Celebrating The Scene That Celebrates Itself; effect heavy, shatteringly loud, sky-sweeping tracks from shoegazers then and now. Follow for more [PIAS] playlists!'. Below the title are buttons for 'PLAY', 'FOLLOWING', and three dots. To the right, it says 'FOLLOWERS 4,257'. A red bracket on the left points to the playlist title, and another red bracket on the left points to the song list.

SONG	ARTIST	ALBUM	DATE	DURATION
+ Only Shallow - Remastered Ver...	My Bloody Valentine	Loveless	2015-03-05	4:18
+ Are You Ready?	Mercury Rev	Are You Ready?	2015-08-06	3:47
+ Sparks	Beach House	Sparks	2015-07-02	5:21
+ Somebody Call a Doctor	Sunflower Bean	Show Me Your Seve...	2015-07-27	3:56
+ Desire	DILLY DALLY	Desire	2015-07-27	3:36
+ Waves	Echo Lake	Era	2015-06-23	6:25
+ Wonderlust	Kid Wave	Wonderlust	2015-06-17	2:55
+ Cannabis	Slim Twig	Cannabis	2015-06-17	5:33
+ ILL	Fews	ILL	2015-06-02	8:17
+ Heroin	Genoah	Heroin	2015-04-27	3:04

Songs in →
playlist are
words

Q&A

- ▶ I'm new to Spotify! What do I do?
- ▶ My friend at the party played Maroon5 using my account. Does that ruin my awesome indie folk playlist?
- ▶ How do I find more music I like?

REFERENCES

Slides by Chris Johnson (worked at Spotify)

https://www.slideshare.net/MrChrisJohnson/from-idea-to-execution-spotifys-discover-weekly/34-Playlist_itself_is_adocumentSongs_inplaylist

<https://www.slideshare.net/MrChrisJohnson/algorithmic-music-recommendations-at-spotify>

Slides by Erik Bernhardsson (worked at Spotify)

https://www.slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818/10-Supervised_collaborative_filtering_is_pretty

Youtube series on recommendation systems by Andrew Ng:

<https://www.youtube.com/watch?v=z0dx-YckFko>

Excellent overview of Spotify's workflow:

<https://hackernoon.com/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe>

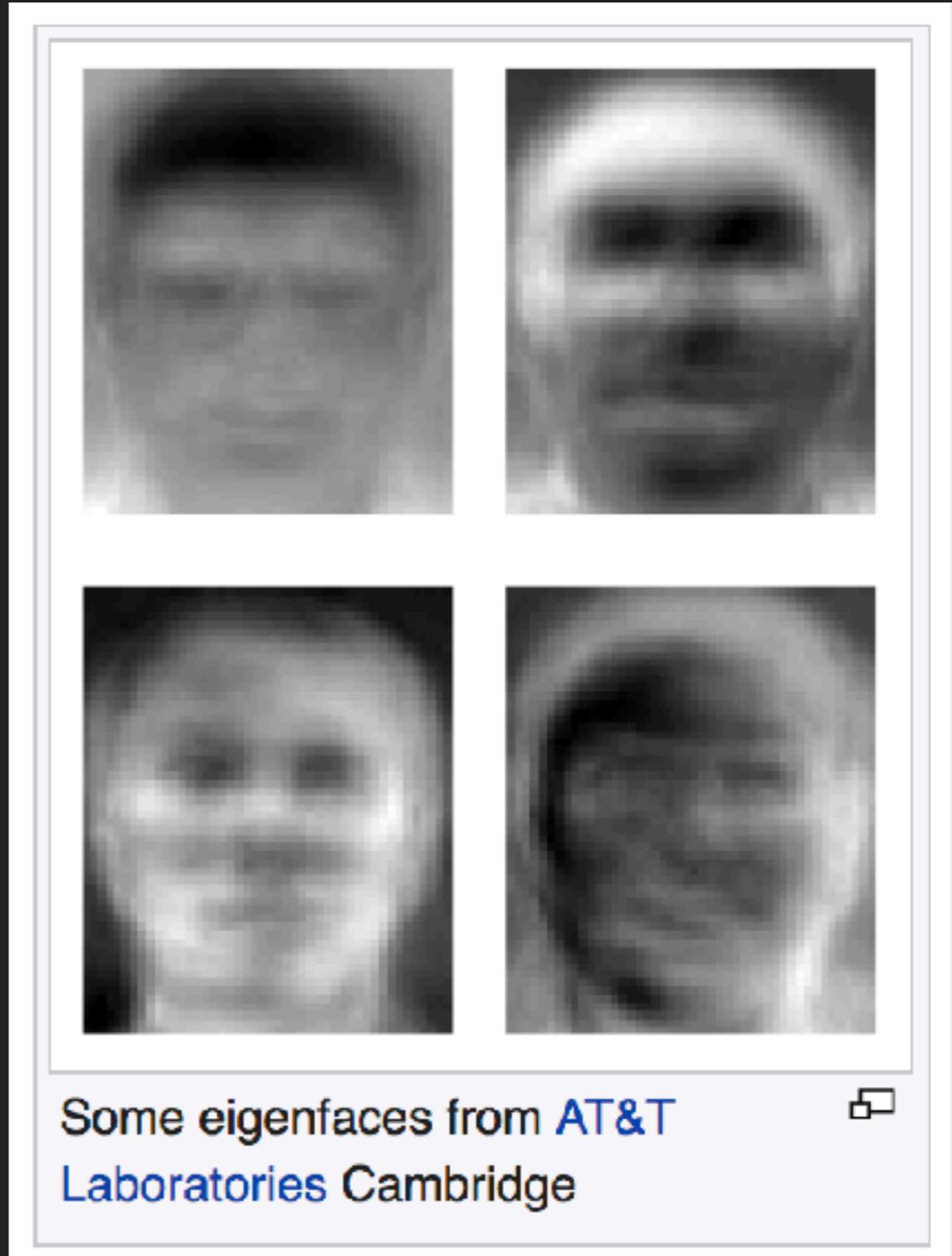
Great talk on matrix factorization by Nicholas Hug:

<https://www.youtube.com/watch?v=9AP-DgFBNP4&t=219s>

REFERENCES

Eigenfaces

<https://en.wikipedia.org/wiki/Eigenface>



Some eigenfaces from [AT&T Laboratories Cambridge](#)

