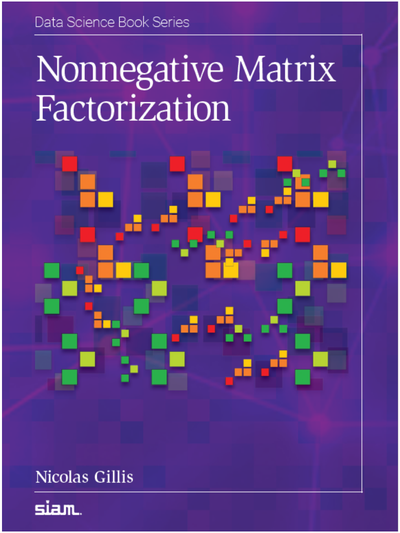
Nonnegative Matrix Factorization



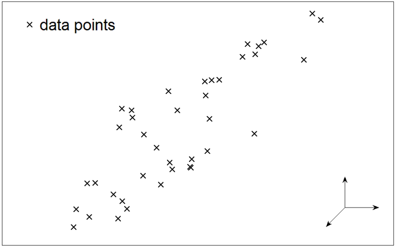
Available at
sites.gogle.com/site/nicolasgillis/book

with codes and additional papers

The setup – Dimensionality reduction for data analysis

- Given a set of n data points m_j $(j = 1, 2, \ldots, n)$, we would like to understand the underlying structure of this data.

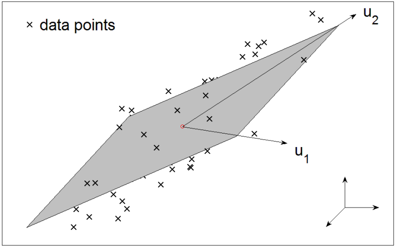
m_j



The setup – Dimensionality reduction for data analysis

- Given a set of n data points m_j ($j = 1, 2, \ldots, n$), we would like to understand the underlying structure of this data.
- A fundamental and powerful tool is linear dimensionality reduction: find a set of r basis vectors u_k ($1 \leq k \leq r$) so that for all j

$$m_j \approx \sum_{k=1}^{r} u_k \, v_{kj}$$

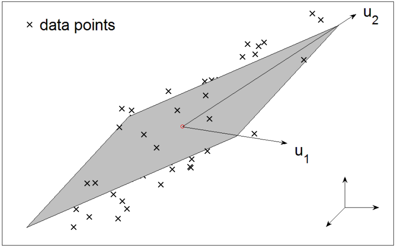


for some weights v_{kj}.

The setup – Dimensionality reduction for data analysis

- Given a set of n data points m_j $(j = 1, 2, \ldots, n)$, we would like to understand the underlying structure of this data.
- A fundamental and powerful tool is linear dimensionality reduction: find a set of r basis vectors u_k $(1 \leq k \leq r)$ so that for all j

$$m_j \approx \sum_{k=1}^{r} u_k \, v_{kj}$$

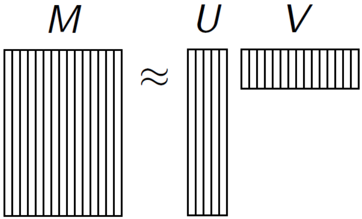


for some weights v_{kj}.

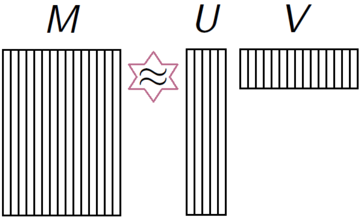
- This is **equivalent** to the low-rank approximation of matrix M:

$$M = [m_1 \, m_2 \, \ldots \, m_n] \approx [u_1 \, u_2 \, \ldots \, u_r] \, [v_1 \, v_2 \, \ldots \, v_n] = UV.$$

Constrained Low-Rank Matrix Approximations



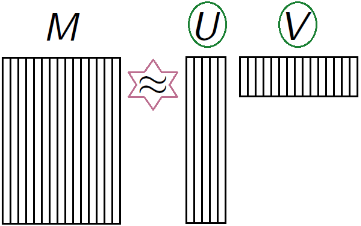
Constrained Low-Rank Matrix Approximations



- How to measure the **error** $||M - UV||$?

  Ex. PCA/truncated SVD use $||X|| = ||X||_F^2 = \sum_{i,j} X_{ij}^2$.

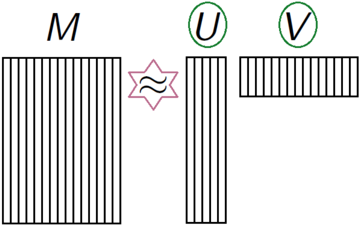
Constrained Low-Rank Matrix Approximations



- How to measure the **error** $||M - UV||$?
  Ex. PCA/truncated SVD use $||X|| = ||X||_F^2 = \sum_{i,j} X_{ij}^2$.

- What **constraints** should the factors $U \in \Omega_U$ and $V \in \Omega_V$ satisfy?
  Ex. PCA has no constraints, k-means a single '1' per column of V.

Constrained Low-Rank Matrix Approximations



- How to measure the **error** $||M - UV||$?
  Ex. PCA/truncated SVD use $||X|| = ||X||_F^2 = \sum_{i,j} X_{ij}^2$.

- What **constraints** should the factors $U \in \Omega_U$ and $V \in \Omega_V$ satisfy?
  Ex. PCA has no constraints, k-means a single '1' per column of V.

Goal of this presentation: show some applications, present several models and discuss some algorithms for **NMF**.

Nonnegative Matrix Factorization (NMF)

Given a matrix $M \in \mathbb{R}_{+}^{p \times n}$ and a factorization rank $r \ll \min(p, n)$, find $U \in \mathbb{R}^{p \times r}$ and $V \in \mathbb{R}^{r \times n}$ such that

$$\min_{U \geq 0, V \geq 0} ||M - UV||_F^2 = \sum_{i,j} (M - UV)_{ij}^2. \tag{NMF}$$

Nonnegative Matrix Factorization (NMF)

Given a matrix $M \in \mathbb{R}_{+}^{p \times n}$ and a factorization rank $r \ll \min(p, n)$, find $U \in \mathbb{R}^{p \times r}$and $V \in \mathbb{R}^{r \times n}$ such that

$$\min_{U \geq 0, V \geq 0} ||M - UV||_F^2 = \sum_{i,j} (M - UV)_{ij}^2. \qquad \text{(NMF)}$$

NMF is a linear dimensionality reduction technique for nonnegative data :

$$\underbrace{M(:, i)}_{\geq 0} \approx \sum_{k=1}^{r} \underbrace{U(:, k)}_{\geq 0} \underbrace{V(k, i)}_{\geq 0} \qquad \text{for all } i.$$

Nonnegative Matrix Factorization (NMF)

Given a matrix $M \in \mathbb{R}_+^{p \times n}$ and a factorization rank $r \ll \min(p, n)$, find $U \in \mathbb{R}^{p \times r}$ and $V \in \mathbb{R}^{r \times n}$ such that

$$\min_{U \geq 0, V \geq 0} ||M - UV||_F^2 = \sum_{i,j} (M - UV)_{ij}^2. \qquad \text{(NMF)}$$

NMF is a linear dimensionality reduction technique for nonnegative data :

$$\underbrace{M(:, i)}_{\geq 0} \approx \sum_{k=1}^{r} \underbrace{U(:, k)}_{\geq 0} \underbrace{V(k, i)}_{\geq 0} \qquad \text{for all } i.$$

Why nonnegativity?

→ **Interpretability**: Nonnegativity constraints lead to easily interpretable factors (and a sparse and part-based representation).

→ **Many applications.** image processing, text mining, hyperspectral unmixing, community detection, clustering, etc.

Application 1: Blind hyperspectral unmixing

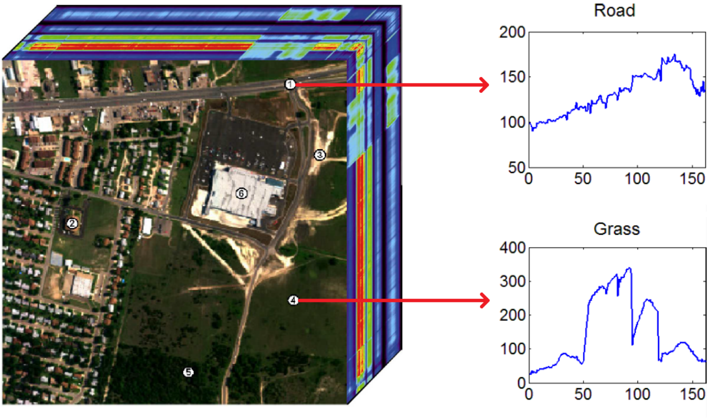


Figure: Urban hyperspectral image, 162 spectral bands and 307-by-307 pixels.

Application 1: Blind hyperspectral unmixing

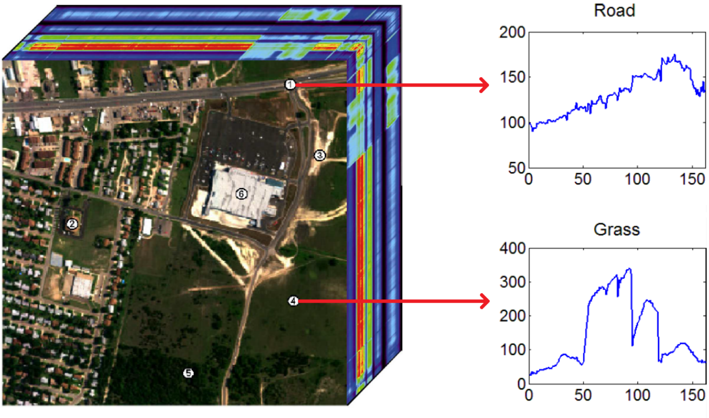
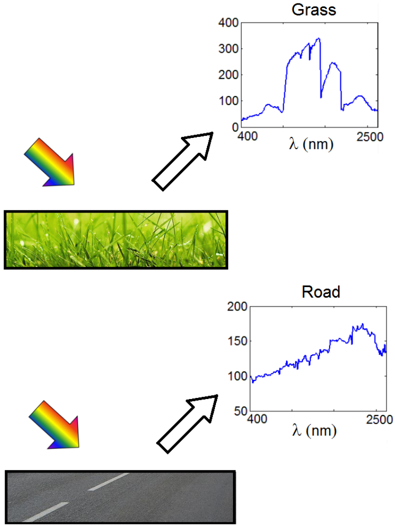
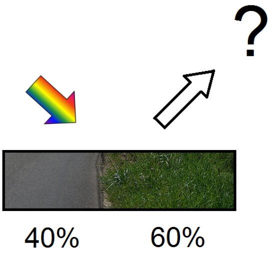


Figure: Urban hyperspectral image, 162 spectral bands and 307-by-307 pixels.

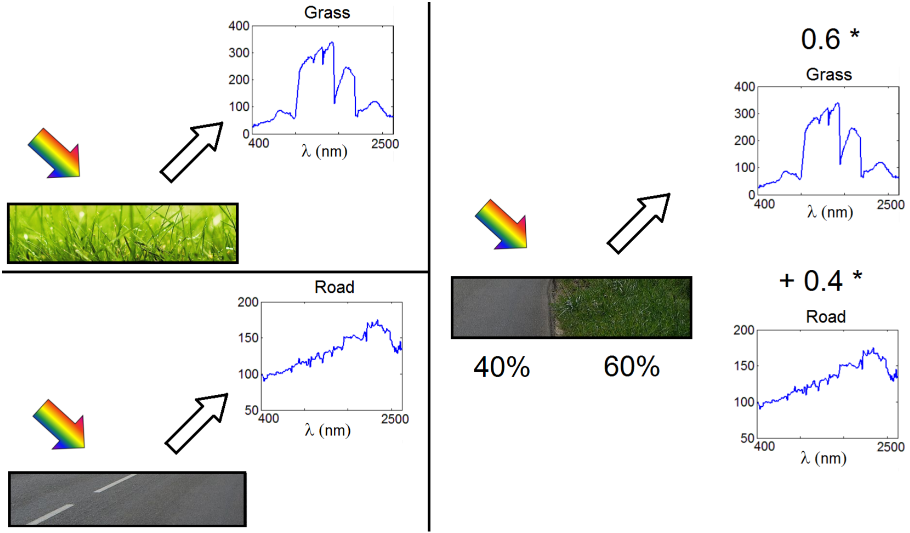
Problem. Identify the materials and classify the pixels.

Linear mixing model

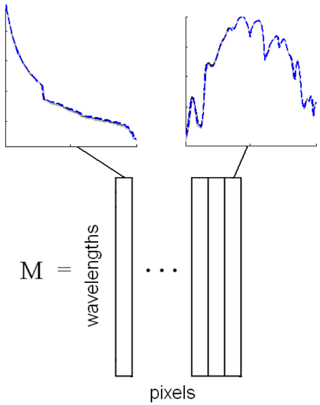




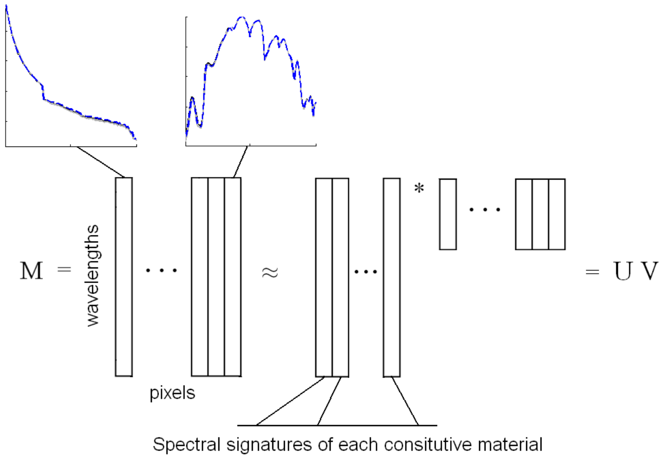
Linear mixing model



Application 1: Blind hyperspectral unmixing with NMF

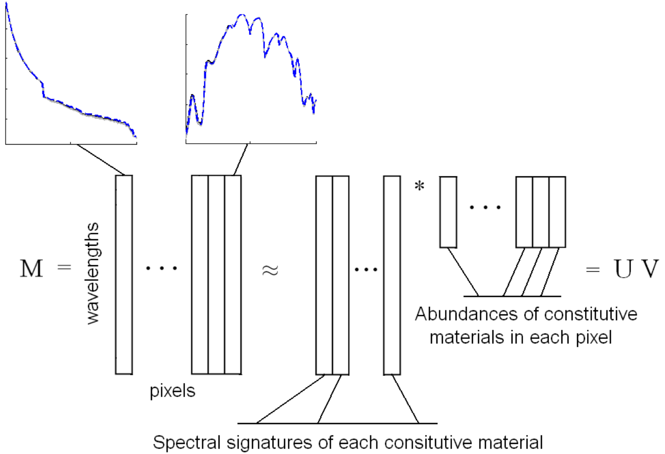


Application 1: Blind hyperspectral unmixing with NMF



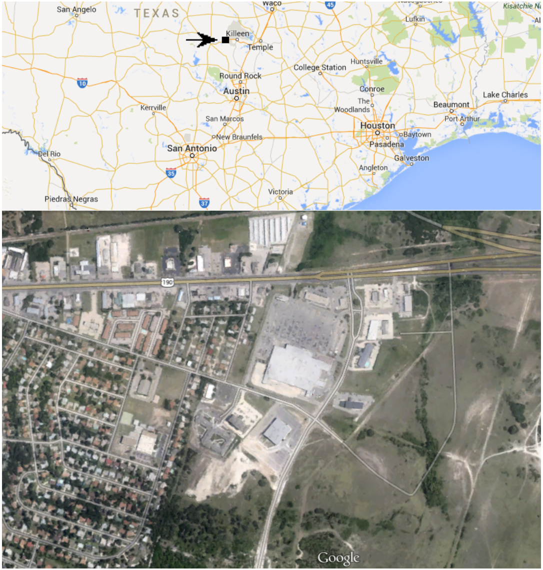
- Basis elements allow to recover the different endmembers: $U \geq 0$;

Application 1: Blind hyperspectral unmixing with NMF



- Basis elements allow to recover the different endmembers: $U \geq 0$;
- Abundances of the endmembers in each pixel: $V \geq 0$.

Urban hyperspectral image



Urban hyperspectral image

$$\underbrace{\mathbf{M}(:,j)}_{\substack{\text{spectral signature}\\ \text{of } j\text{th pixel}}} \approx \sum_{k=1} \underbrace{\mathbf{U}(:,k)} \underbrace{\mathbf{V}(k,j)} .$$

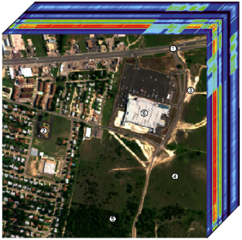
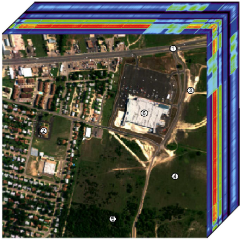


Figure: Decomposition of the Urban dataset.

Urban hyperspectral image

$$\underbrace{\mathrm{M}(:,j)}_{\substack{\text{spectral signature}\\ \text{of } j\text{th pixel}}} \approx \sum_{k=1} \underbrace{\mathrm{U}(:,k)}_{\substack{\text{spectral signature}\\ \text{of } k\text{th endmember}}} \underbrace{\mathrm{V}(k,j)} .$$



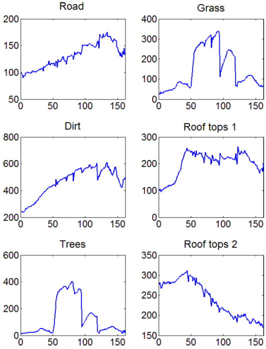


Figure: Decomposition of the Urban dataset.

Urban hyperspectral image

$$\underbrace{\mathrm{M}(:,j)}_{\substack{\text{spectral signature}\\ \text{of } j\text{th pixel}}} \approx \sum_{k=1} \underbrace{\mathrm{U}(:,k)}_{\substack{\text{spectral signature}\\ \text{of } k\text{th endmember}}} \underbrace{\mathrm{V}(k,j)}_{\substack{\text{abundance of } k\text{th endmember}\\ \text{in } j\text{th pixel}}} .$$

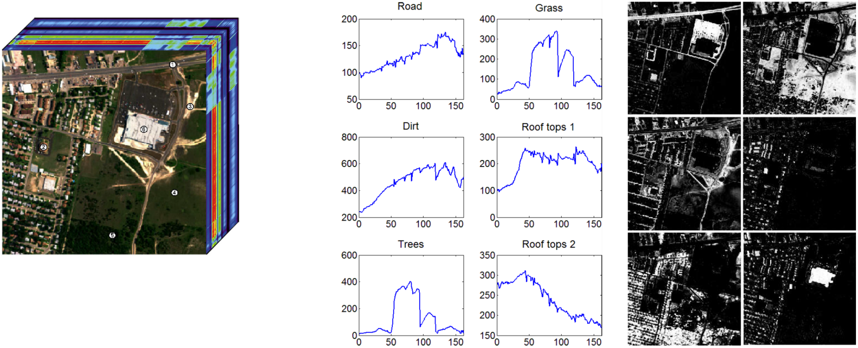
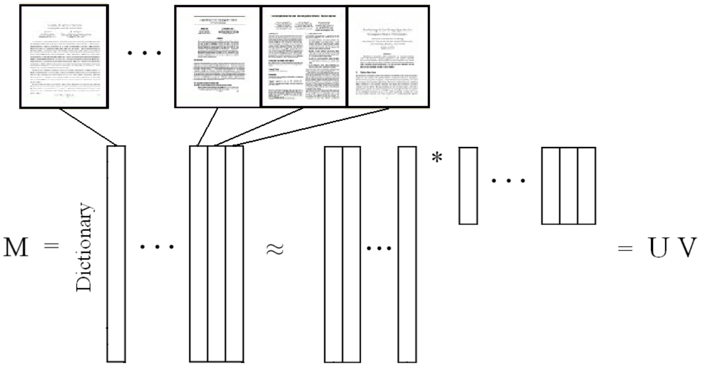
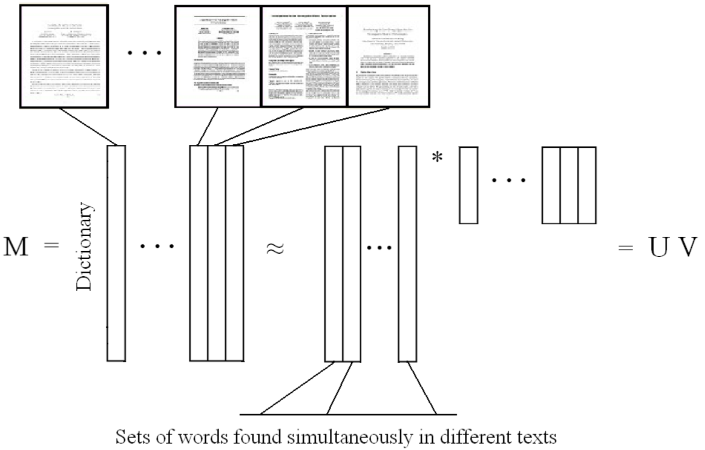


Figure: Decomposition of the Urban dataset.

Application 2: topic recovery and document classification

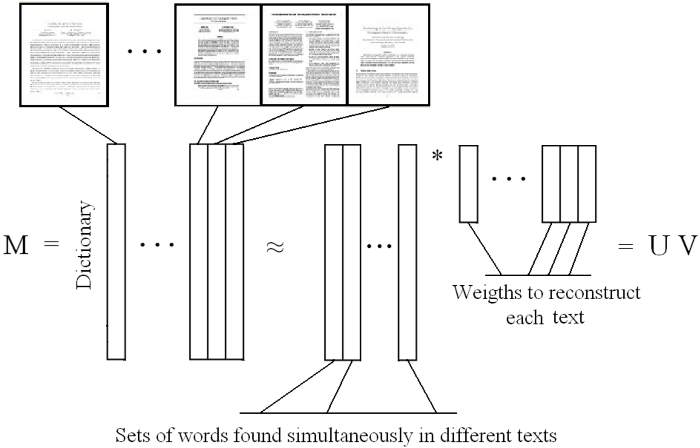


Application 2: topic recovery and document classification



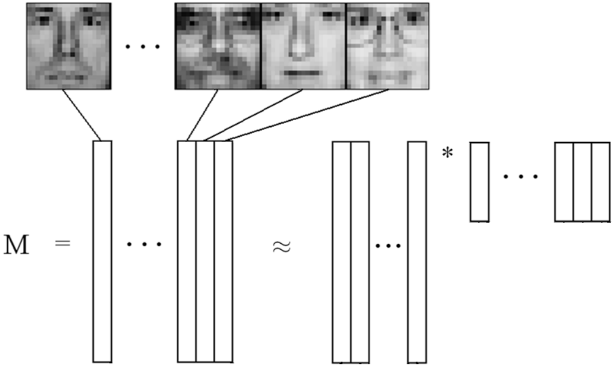
- Basis elements allow to recover the different topics;

Application 2: topic recovery and document classification

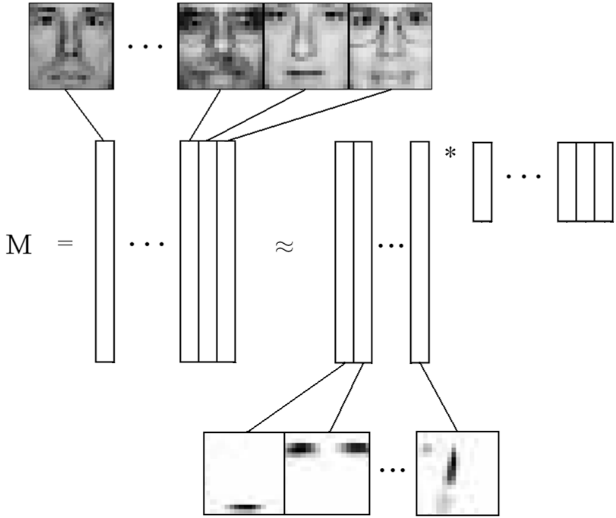


- Basis elements allow to recover the different topics;
- Weights allow to assign each text to its corresponding topics.

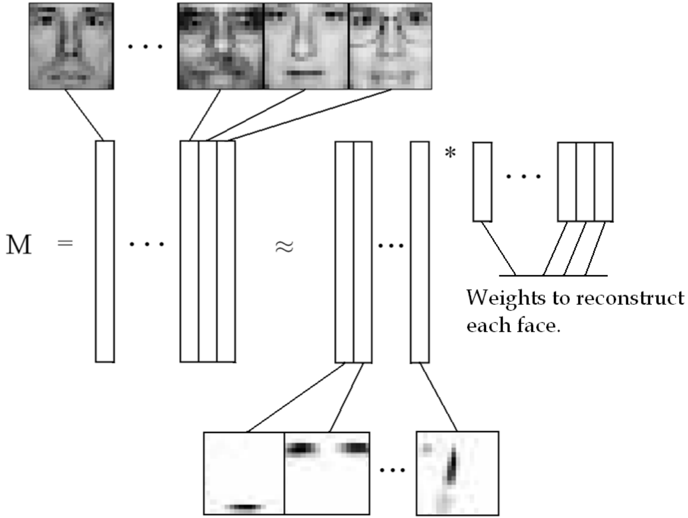
Application 3: feature extraction and classification



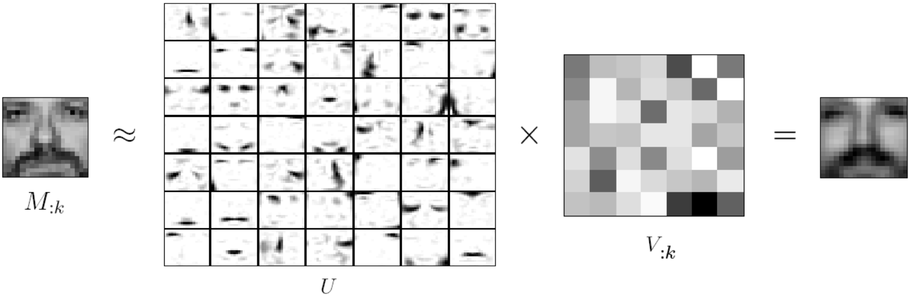
Application 3: feature extraction and classification



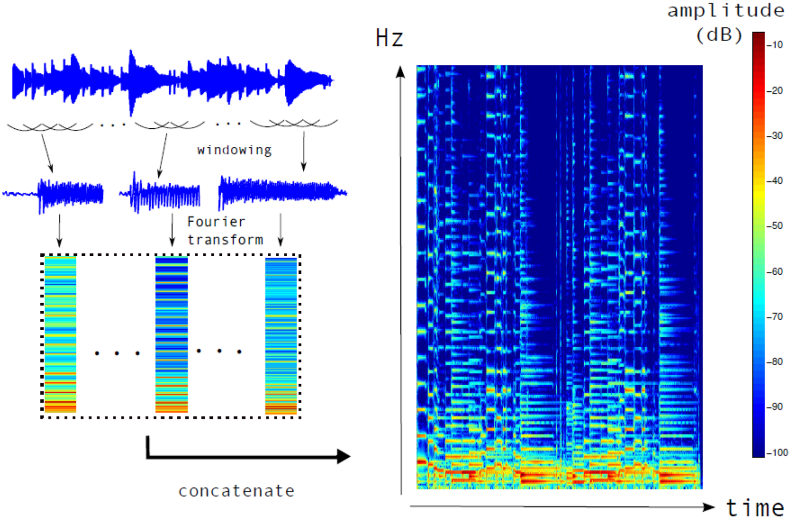
Application 3: feature extraction and classification



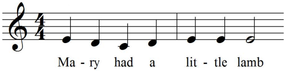
The basis elements extract facial features such as eyes, nose and lips.

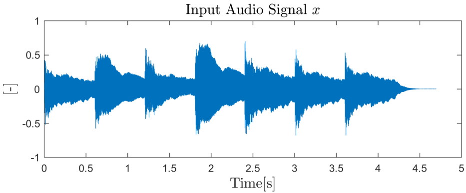

$M_{:k}$
$\approx$
U
$\times$
$V_{:k}$
$=$

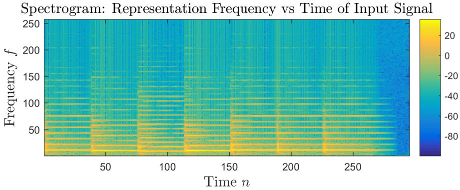
Application 4: audio source separation



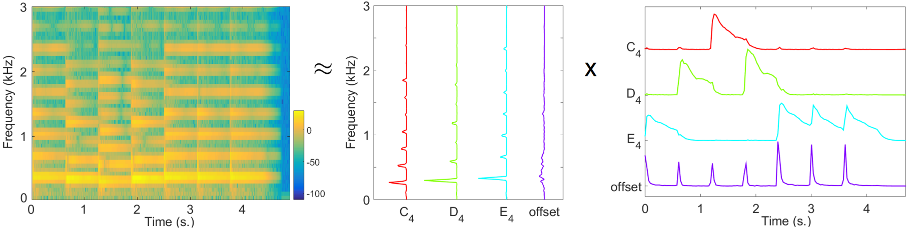
Application 4: audio source separation







Application 4: audio source separation



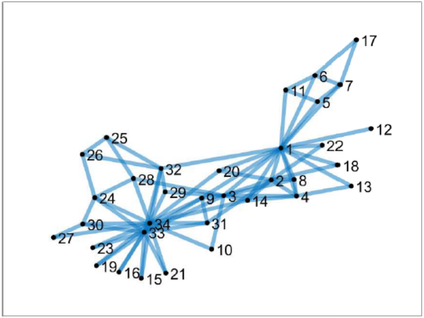
doudou_melody.webm

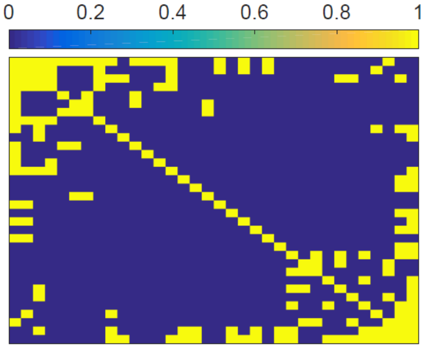
Application 5: community detection

$M_{i,j} = \exp\left(-c\|x_i - x_j\|^2\right)$ is an entrywise positive and PSD matrix.
Consider the symmetric NMF model $M \approx UU^\top$ where $U \geq 0$.

Application 5: community detection

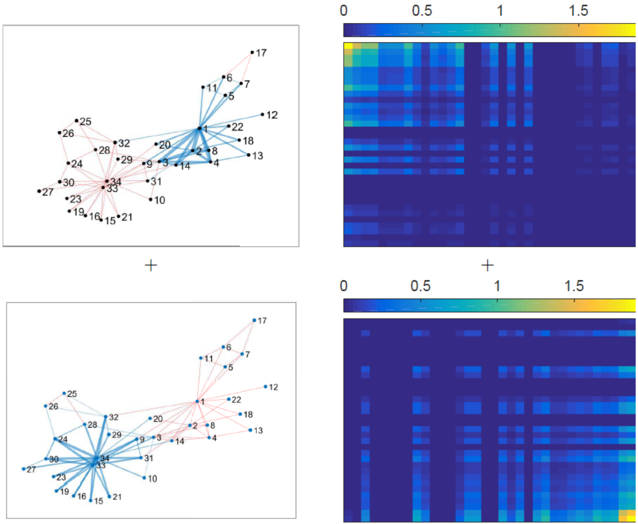
$M_{i,j} = \exp\left(-c\|x_i - x_j\|^2\right)$ is an entrywise positive and PSD matrix.
Consider the symmetric NMF model $M \approx UU^\top$ where $U \geq 0$.





Application 5: community detection

$M_{i,j} = \exp\left(-c\|x_i - x_j\|^2\right)$ is an entrywise positive and PSD matrix.
Consider the symmetric NMF model $M \approx UU^\top$ where $U \geq 0$.



Application 6: recommender systems

In some cases, some entries are missing/unknown.

For example, we would like to predict how much someone is going to like a movie based on its movie preferences (e.g., 1 to 5 stars) :

$$
\begin{array}{cc}
 & \text{Users} \\
\text{Movies} & \begin{bmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{bmatrix}
\end{array}
$$

Application 6: recommender systems

In some cases, some entries are missing/unknown.

For example, we would like to predict how much someone is going to like a movie based on its movie preferences (e.g., 1 to 5 stars) :

$$
\begin{array}{cc}
 & \text{Users} \\
\text{Movies} & \begin{bmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{bmatrix}
\end{array}
$$

Huge potential in electronic commercial sites (movies, books, music, . . .). Good recommendations will increase the propensity of a purchase.

Low-rank matrix approximations

The behavior of users is modeled using linear combination of 'feature' users (related to age, sex, culture, etc.)

$$\underbrace{M(:,j)}_{\text{user } j} \approx \sum_{k=1}^{r} \underbrace{U(:,k)}_{\text{feature user } k} \underbrace{V(k,j)}_{\text{weights}}$$

Low-rank matrix approximations

The behavior of users is modeled using linear combination of 'feature' users (related to age, sex, culture, etc.)

$$\underbrace{M(:,j)}_{\text{user } j} \approx \sum_{k=1}^{r} \underbrace{U(:,k)}_{\text{feature user } k} \underbrace{V(k,j)}_{\text{weights}}$$

Or equivalently, movies ratings are modeled as linear combinations of 'feature' movies (related to the genres - child oriented, serious vs. escapist, thriller, romantic, actors, etc.).

$$\underbrace{M(i,:)}_{\text{movie } i} \approx \sum_{k=1}^{r} \underbrace{U(i,k)}_{\text{weights}} \underbrace{V(k,:)}_{\text{genre } k}$$

For example, using a rank-2 factorization on the Netflix dataset, female vs. male and serious vs. escapist behaviors were extracted.

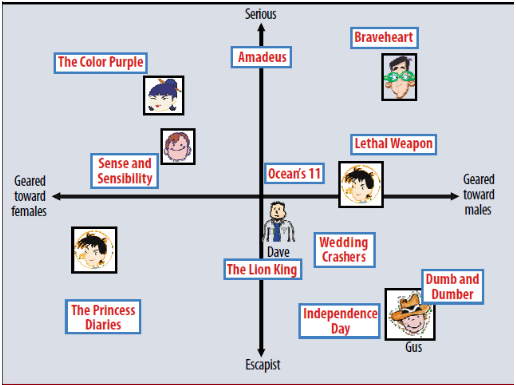


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Koren, Bell, Volinsky, *Matrix Factorization Techniques for Recommender Systems, 2009*.
Winners of the Netflix prize 1,000,000$.

NMF is easily interpretable

$$X = \begin{pmatrix} 2 & 3 & 2 & ? & ? \\ ? & 1 & ? & 3 & 2 \\ 1 & ? & 4 & 1 & ? \\ 5 & 4 & ? & 3 & 2 \\ ? & 1 & 2 & ? & 4 \\ 1 & ? & 3 & 4 & 3 \end{pmatrix} \approx \begin{pmatrix} 1.6 & 0.9 & 2.2 \\ 0.9 & 2.3 & 0.4 \\ 0.2 & 0.8 & 5.0 \\ 5.0 & 0.8 & 0.4 \\ 1.4 & 5.0 & 0.0 \\ 0.4 & 3.3 & 2.3 \end{pmatrix} \begin{pmatrix} 1.0 & 0.7 & 0.0 & 0.4 & 0.3 \\ 0.1 & 0.0 & 0.4 & 1.1 & 0.7 \\ 0.2 & 0.8 & 0.7 & 0.0 & 0.2 \end{pmatrix}$$

$$= \begin{pmatrix} 2.0 & 3.0 & 2.0 & 1.7 & 1.5 \\ 1.1 & 1.0 & 1.2 & 3.0 & 2.0 \\ 1.0 & 4.2 & 4.0 & 1.0 & 1.6 \\ 5.0 & 4.0 & 0.6 & 3.0 & 2.0 \\ 1.6 & 1.0 & 2.0 & 6.3 & 4.0 \\ 1.0 & 2.2 & 3.0 & 4.0 & 3.0 \end{pmatrix},$$

Application 7: community detection

Dataset of 101 animals with 17 characteristics, including:

	hair	feathers	eggs	aquatic	milk
bass	0	0	1	1	0
bear	1	0	0	0	1
chicken	0	1	1	0	0
gorilla	1	0	0	0	1
ostrich	0	1	1	0	0
seahorse	0	0	1	1	0

Example from `http://archive.ics.uci.edu/dataset/111/zoo`.

Application 7: community detection

Dataset of 101 animals with 17 characteristics, including:

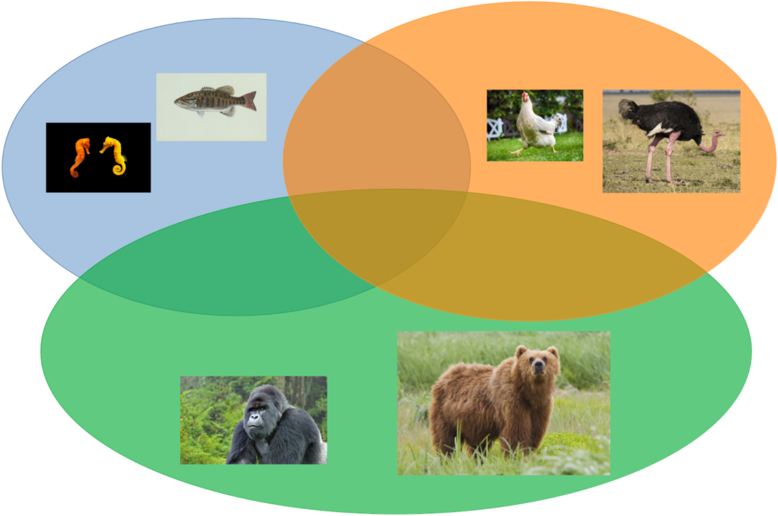
	hair	feathers	eggs	aquatic	milk
bass	0	0	1	1	0
bear	1	0	0	0	1
chicken	0	1	1	0	0
gorilla	1	0	0	0	1
ostrich	0	1	1	0	0
seahorse	0	0	1	1	0

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Example from `http://archive.ics.uci.edu/dataset/111/zoo`.

Application 7: community detection

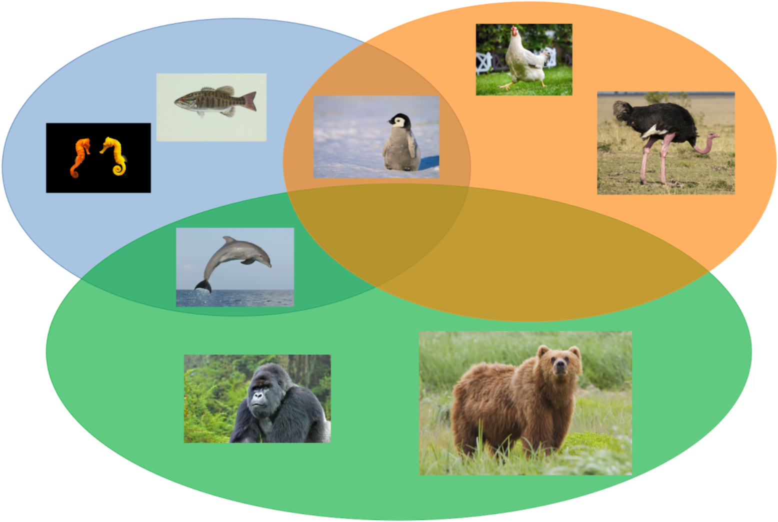
Dataset of 101 animals with 17 characteristics, including:



Example from `http://archive.ics.uci.edu/dataset/111/zoo`.

Application 7: community detection

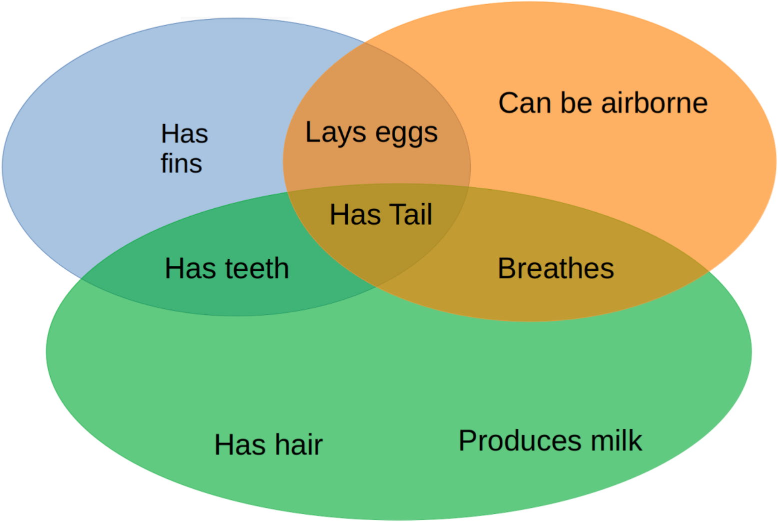
Dataset of 101 animals with 17 characteristics, including:



Example from `http://archive.ics.uci.edu/dataset/111/zoo`.

Application 7: community detection

Dataset of 101 animals with 17 characteristics, including:



Example from `http://archive.ics.uci.edu/dataset/111/zoo`.