

# Non-negative Matrix Factorization (NMF)



# Problem Motivation

- Soft clustering
  - Clustering where each observation can have partial membership in each cluster
- Identify latent features
  - Identify topics in a corpus of text
  - Identify parts of faces in facial recognition

# Example: Baking a Cake

- Given a cake, we can directly measure carbohydrates, fat, and protein
- We *cannot* directly observe how much flour and sugar went into the cake
- NMF can learn ingredients from the final product
  - NMF learns the ingredients and the amount of each ingredient in each cake

# Example: Baking a Cake

- Recipe is a linear combination of ingredients
  - 2 cups flour
  - 1 cup sugar

# Example: Baking a Cake

- Ingredients are linear combination of carbohydrates, fat, and protein
  - **Flour:**  
20g carbohydrates, 5g protein, 1g fat = **[20, 5, 1]**
  - **Sugar:**  
10g carbohydrates, 0g protein, 1g fat = **[10, 0, 1]**

# Example: Baking a Cake

Recipe and ingredients can be represented by vectors/matrices

$$\text{Recipe} = [\text{flour}, \text{sugar}] = [2, 1]$$

$$\begin{aligned}\text{Ingredients} &= \begin{bmatrix} f_c & f_p & f_f \\ s_c & s_p & s_f \end{bmatrix} \\ &= \begin{bmatrix} 20 & 5 & 1 \\ 10 & 0 & 1 \end{bmatrix}\end{aligned}$$

# Example: Baking a Cake

Cake is product of recipe vector and ingredient matrix

$$\text{Cake} = (\text{recipe})[\text{Ingredients}]$$

$$= \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{bmatrix} 20 & 5 & 1 \\ 10 & 0 & 1 \end{bmatrix}$$

$$= 2 \begin{pmatrix} 20 & 5 & 1 \end{pmatrix} + 1 \begin{pmatrix} 10 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 50 & 10 & 3 \end{pmatrix}$$

# Example: Baking a Cake

Many cakes can be represented by product of 2 matrices

$$\begin{bmatrix} c_1 & p_1 & f_1 \\ c_2 & p_2 & f_2 \\ c_3 & p_3 & f_3 \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} w_{1f} & w_{1s} \\ w_{2f} & w_{2s} \\ w_{3f} & w_{3s} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} f_c & f_p & f_f \\ s_c & s_p & s_f \end{bmatrix}$$



# Example: Unbaking a Cake

*Finding the recipe*

Given cakes and a set of ingredients, can we figure out the recipe?

$$\begin{bmatrix} 30 & 5 & 2 \\ 20 & 5 & 3 \\ 25 & 3 & 3 \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} w_{1f} & w_{1s} \\ w_{2f} & w_{2s} \\ w_{3f} & w_{3s} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} 20 & 5 & 1 \\ 10 & 0 & 1 \end{bmatrix}$$

# Example: Unbaking a Cake

*Finding the recipe*

Can perform OLS minimization to find the recipe

$$\begin{bmatrix} 25 & 3 & 3 \end{bmatrix} = (w_f \quad w_s) \begin{bmatrix} 20 & 5 & 1 \\ 10 & 0 & 1 \end{bmatrix}$$

$$25 \approx \hat{c} = 20w_{3f} + 10w_{3s}$$

$$3 \approx \hat{p} = 5w_{3f} + 0w_{3s}$$

$$3 \approx \hat{f} = 1w_{3f} + 1w_{3s}$$

$$\operatorname{argmin}_{\vec{w}} \sum_{y \in (c,p,f)} (y - \hat{y})^2$$

# Example: Unbaking a Cake

*Finding the ingredients*

Given cakes and a set of recipes, can we figure out the ingredients?

$$\begin{bmatrix} 30 & 5 & 2 \\ 20 & 5 & 3 \\ 25 & 3 & 3 \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} f_c & f_p & f_f \\ s_c & s_p & s_f \end{bmatrix}$$

# Example: Unbaking a Cake

*Finding the ingredients*

Can perform OLS minimization to find the ingredients

$$\begin{bmatrix} 25 & 3 & 3 \end{bmatrix} = (1 \quad 2) \begin{bmatrix} f_c & f_p & f_f \\ s_c & s_p & s_f \end{bmatrix}$$

$$25 = \hat{c} \approx 1f_c + 2s_c = 1f_c + 2s_c + 0f_p + 0s_p + 0f_f + 0s_f$$

$$3 = \hat{p} \approx 1f_p + 2s_p = 0f_c + 0s_c + 1f_p + 2s_p + 0f_f + 0s_f$$

$$3 = \hat{f} \approx 1f_f + 2s_f = 0f_c + 0s_c + 0f_p + 0s_p + 1f_f + 2s_f$$

...continued for all observations

# Example: Unbaking a Cake

*Finding the ingredients and the recipes*

What if you don't know the recipe proportions *or* the ingredients?

$$\begin{bmatrix} 30 & 5 & 2 \\ 20 & 5 & 3 \\ 25 & 3 & 3 \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} w_{1f} & w_{1s} \\ w_{2f} & w_{2s} \\ w_{3f} & w_{3s} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} f_c & f_p & f_f \\ s_c & s_p & s_f \end{bmatrix}$$

# Alternating Least Squares

$$X = WH$$

1. Initialize\* W and H
2. Solve for H holding W constant (minimize OLS)
3. Solve W holding H constant (minimize OLS)
4. Repeat steps 2 and 3 until convergence

*\*Initialization methods vary. Random initialization and k-means are common.*

# Interpreting NMF Results

$$X = WH$$

$X$  = (observations) x (features)

$W$  = (observations) x (latent features)

$H$  = (latent features) x (observed features)

# Interpreting NMF Results

## Topic Modeling

$X = (\text{documents}) \times (\text{word counts})$

$W = (\text{documents}) \times (?)$

$H = (?) \times (\text{word counts})$

→ Latent features correspond to topics. This is explored in today's assignment.



# Interpreting NMF Results

## Face Identification

Suppose  $X$  is a set of images of faces (all the same size)

$X = (\text{images}) \times (\text{pixel intensities})$

$W = (\text{image}) \times (?)$

$H = (?) \times (\text{pixel intensities})$

→ Latent features correspond to parts of faces

# Interpreting NMF Results

## Face Identification

Basis images from NMF on faces <http://www.cs.ucsb.edu/~mturk/pubs/IJPRAI05.pdf>



# Extensions

- Regularization
  - Can add lasso or ridge term to ALS
- Cross-validation
  - See “Rank Selection in Low-rank Matrix Approximations: A Study of Cross-Validation for NMFs” <https://www.cs.umd.edu/~bhargav/nips2010.pdf>
- Other cost functions
- Other optimization methods
  - Multiplicative update rule