

k-means

Lecture 12

Sibylle Hess and Mahdi Shafiee

March, 2021

Last Lecture..

Truncated SVD solves the Rank- r Matrix Factorization Problem

The Rank- r Matrix Factorization Problem

Given: a data matrix $D \in \mathbb{R}^{n \times d}$ and a rank $r < \min\{n, d\}$.

Find: matrices $X \in \mathbb{R}^{d \times r}$ and $Y \in \mathbb{R}^{n \times r}$ whose product approximates the data matrix:

$$\min_{X,Y} \|D - YX^\top\|^2 \quad \text{s.t. } X \in \mathbb{R}^{d \times r}, Y \in \mathbb{R}^{n \times r}$$

Solution: Let $D = U\Sigma V^\top$ be the SVD of D . Choose $X \in \mathbb{R}^{d \times r}$ and $Y \in \mathbb{R}^{n \times r}$ such that

$$YX^\top = U_{\cdot\mathcal{R}} \Sigma_{\mathcal{R}\mathcal{R}} V_{\cdot\mathcal{R}}^\top$$

Truncated SVD

The approximation $D \approx U_{\mathcal{R}} \Sigma_{\mathcal{R} \mathcal{R}} V_{\mathcal{R}}^T$ is called **truncated SVD**.

$$D \approx n \left\{ \begin{array}{c|cc} & U_{\cdot 1} & \cdots & U_{\cdot r} \\ \hline & \sigma_1 & & \\ & \vdots & & \sigma_r \\ & V_{\cdot 1}^T & & \\ & \vdots & & V_{\cdot r}^T \\ & \hline & d & \end{array} \right.$$

The diagram illustrates the truncated SVD approximation $D \approx U_{\mathcal{R}} \Sigma_{\mathcal{R} \mathcal{R}} V_{\mathcal{R}}^T$. It shows a large matrix D approximated by a sum of three matrices. The first matrix is $U_{\cdot 1} \cdots U_{\cdot r}$, where each column $U_{\cdot i}$ is shown as a vertical vector. The second matrix is a diagonal matrix $\sigma_1 \dots \sigma_r$, where each diagonal element σ_i is shown as a vertical vector. The third matrix is $V_{\cdot 1}^T \dots V_{\cdot r}^T$, where each row $V_{\cdot i}^T$ is shown as a horizontal vector. A brace under the first matrix indicates its width is r , and a brace under the third matrix indicates its height is d .

Truncated SVD

Theorem (MF is Nonconvex)

The rank- r matrix factorization problem, defined for a matrix $D \in \mathbb{R}^{n \times d}$ and a rank $r < \min\{n, d\}$ as

$$\min_{X, Y} \|D - YX^\top\|^2 \quad s.t. \quad X \in \mathbb{R}^{d \times r}, Y \in \mathbb{R}^{n \times r}$$

*is a **nonconvex** optimization problem.*

Matrix Completion for Recommender Systems

		Movies			
		A	B	C	D
Users	1	★★★★★	?	★★★★★	★★★★★
	2	?	★★★★★	★★★★★	?
	3	★★★★★	★★★★★	★★★★★	★★★★★
	4	★★★★★	?	★★★★★	★★★★★
	5	★★★★★	★★★★★	?	?
	6	?	★★★★★	★★★★★	★★★★★

Can we fill the ? with the rating which would be given by the user if (s)he had seen the movie?

SVD in the Scope of Movie Recommender Systems

$$\begin{pmatrix} 5 & \mu & 1 & 1 \\ \mu & 1 & 5 & \mu \\ 2 & 1 & 5 & 3 \\ 4 & \mu & 4 & 2 \\ 5 & 5 & \mu & 1 \\ \mu & 1 & 5 & 3 \end{pmatrix} \approx \begin{pmatrix} -0.3 & 0.5 \\ -0.4 & -0.4 \\ -0.4 & -0.4 \\ -0.4 & 0.1 \\ -0.5 & 0.5 \\ -0.4 & -0.4 \end{pmatrix} \begin{pmatrix} -9.0 & -5.8 & -9.5 & -5.3 \\ 2.6 & 3.3 & -3.3 & -2.2 \end{pmatrix}$$

Every user's preferences are approximated by a linear combination of the rows in the second matrix:

$$(5 \quad \mu \quad 1 \quad 1) \approx -0.3 \cdot (-9.0 \quad -5.8 \quad -9.5 \quad -5.3) \\ + 0.5 \cdot (2.6 \quad 3.3 \quad -3.3 \quad -2.2)$$

Today: k -means

1

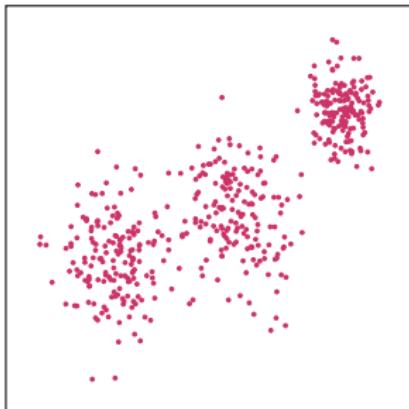
Informal Problem Description

Clustering Means to Group Data Points According to a Similarity Criterion



Question: what are clusters in a deck of cards?

Clustering is a Task with Multiple Valid Outcomes



- 1 How many clusters do we have?
- 2 Do they overlap?
- 3 How are clusters characterized?

Cluster models differ according to the answers to these questions.

2

Derive the Formal Problem Definition

The Cluster Model of k -means

- 1 How many clusters do we have? Let the user decide..
- 2 Do they overlap? No. Every point belongs to exactly one cluster

$$\mathcal{C}_s \cap \mathcal{C}_t = \emptyset, \quad \mathcal{C}_1 \cup \dots \cup \mathcal{C}_r = \{1, \dots, n\}$$

That is, $\{\mathcal{C}_1, \dots, \mathcal{C}_r\}$ is a partition of $\{1, \dots, n\}$. We denote the set of all partitions from $\{1, \dots, n\}$ with \mathcal{P}_n .

- 3 How are clusters characterized? Points within a cluster are close in average:

$$\frac{1}{|\mathcal{C}_s|} \sum_{i,j \in \mathcal{C}_s} \|D_{i\cdot} - D_{j\cdot}\|^2 \text{ is small.}$$

The k -means Objective

Given: a data matrix $D \in \mathbb{R}^{n \times d}$ and the number of clusters r .

Find: clusters $\{\mathcal{C}_1, \dots, \mathcal{C}_r\} \in \mathcal{P}_n$ which create a partition of $\{1, \dots, n\}$, minimizing the distance between points within clusters (within cluster scatter):

$$\min_{\{\mathcal{C}_1, \dots, \mathcal{C}_r\} \in \mathcal{P}_n} Dist(\mathcal{C}_1, \dots, \mathcal{C}_r) = \sum_{s=1}^r \frac{1}{|\mathcal{C}_s|} \sum_{j, i \in \mathcal{C}_s} \|D_{j \cdot} - D_{i \cdot}\|^2 \quad (1)$$

3

Optimization

Ok, we have here now one problem:

The standard optimization methods relying on gradients do not apply, this is a **discrete optimization problem.**

How can we optimize the objective of k -means when the gradients are not defined?

Transform the objective to get a better idea.

Minimizing the Within Cluster Distance Means Minimizing the Distance of Points to their Centroid

Theorem (*k*-means centroid objective)

The k-means objective in Eq. (1) is equivalent to

$$\min_{s=1}^r \sum_{i \in \mathcal{C}_s} \|D_{i \cdot} - X_{\cdot s}^\top\|^2 \quad \text{s.t. } X_{\cdot s} = \frac{1}{|\mathcal{C}_s|} \sum_{i \in \mathcal{C}_s} D_{i \cdot}^\top, \\ \{\mathcal{C}_1, \dots, \mathcal{C}_r\} \in \mathcal{P}_n$$

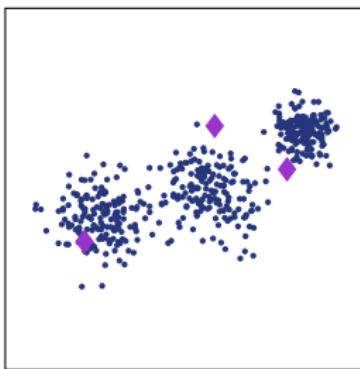
$X_{\cdot s}$ is the **centroid** (the arithmetic mean position) of all points assigned to cluster \mathcal{C}_s .

Does this notion of centroids
deliver more easily solvable
sub-problems?

Maybe it's more easy to
compute the centroids given
the clusters and vice versa
instead of computing clusters
and centroids simultaneously?

Minimizing the Distance of Points to their Centroids

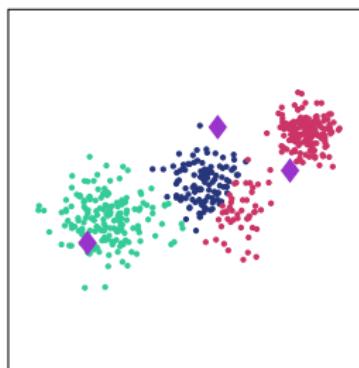
Let us start with some randomly sampled centroids (the purple diamonds).



Question: how do we assign points to clusters?

Minimizing the Distance of Points to their Centroids

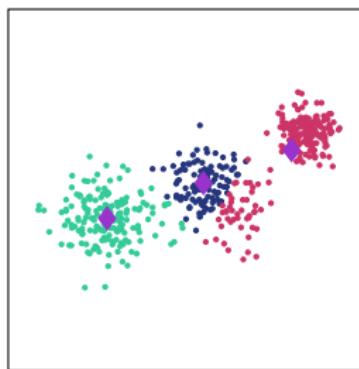
We assign every point to the cluster with the closest centroid.



Problem: now the centroids are not actually centroids of all points in one cluster!

Minimizing the Distance of Points to their Centroids

We update the centroids.



Observation: now we can again decrease the objective function by assigning the points to their closest centroid!

Congratulations!

You just came up with the algorithm for k -means on your own.

Lloyds k -means Algorithm

```
1: function K-MEANS( $r, D$ )
2:    $X \leftarrow \text{INITCENTROIDS}(D, r)$             $\triangleright$  centroid initialization
3:   while not converged do
4:     for  $s \in \{1, \dots, r\}$  do
5:        $\mathcal{C}_s \leftarrow \left\{ i \middle| s = \arg \min_{1 \leq t \leq r} \left\{ \|X_{\cdot t} - D_{i \cdot}^{\top}\|^2 \right\}, 1 \leq i \leq n \right\}$ 
6:     end for
7:     for  $s \in \{1, \dots, r\}$  do
8:        $X_{\cdot s} \leftarrow \frac{1}{|\mathcal{C}_s|} \sum_{i \in \mathcal{C}_s} D_{i \cdot}^{\top}$             $\triangleright$  centroid update
9:     end for
10:   end while
11:   return  $\{\mathcal{C}_1, \dots, \mathcal{C}_r\}$ 
12: end function
```

Cool, we have now **a** algorithm
for the discrete optimization
problem of k -means.

How good is this algorithm?

Does it **converge**? Is it just a heuristic or can we derive some quality guarantees of the result?

These questions can all be answered under the more general framework of **matrix factorization**.

Indicating Clusters by a Binary Matrix

Let $Y \in \{0, 1\}^{n \times r}$ such that $Y_{is} = 1$ if and only if $i \in \mathcal{C}_s$.

Every point belongs to exactly one cluster if and only if

$$|Y_{i\cdot}| = 1 \text{ for all } i \in \{1, \dots, n\},$$

We denote with $\mathbb{1}^{n \times r}$ the set of all **binary matrices** which indicate **a partition** of n points into r sets:

$$\mathbb{1}^{n \times r} = \{Y \in \{0, 1\}^{n \times r} \mid |Y_{i\cdot}| = 1 \text{ for } i \in \{1, \dots, n\}\}$$

The Centroid Matrix

Given a cluster indicator matrix $Y \in \mathbb{1}^{n \times r}$, the s th centroid is

$$X_{\cdot s} = \frac{1}{|\mathcal{C}_s|} \sum_{i \in \mathcal{C}_s} D_{i \cdot}^T = \frac{1}{|Y_{\cdot s}|} \sum_{i=1}^n Y_{is} D_{i \cdot}^T = \frac{1}{|Y_{\cdot s}|} D^T Y_{\cdot s}.$$

We can compute the matrix X which gathers all **centroids** column-wise by

$$X = D^T Y \begin{pmatrix} \frac{1}{|Y_{\cdot 1}|} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{|Y_{\cdot r}|} \end{pmatrix} = D^T Y (Y^T Y)^{-1}.$$

k-means is Matrix Factorization

Theorem (k-means MF objective)

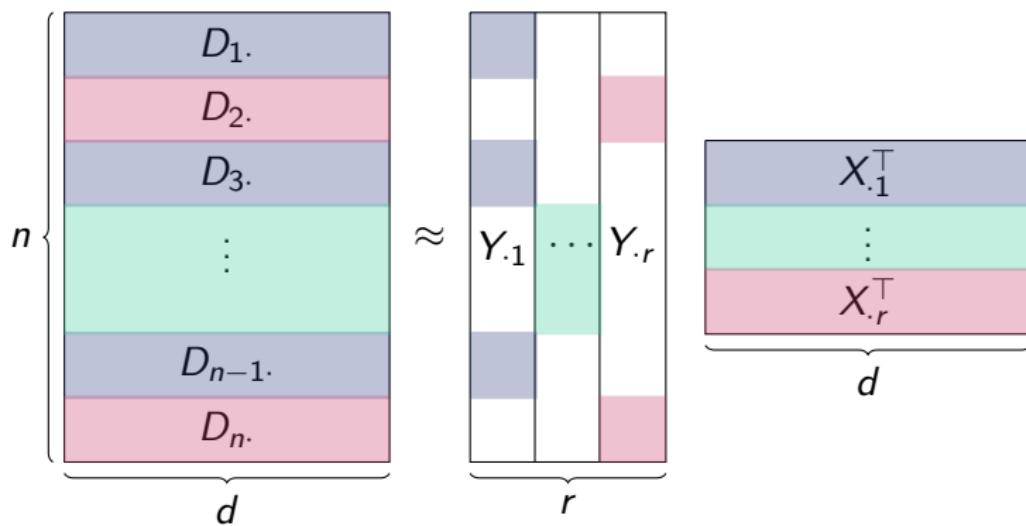
The k-means objective in Eq. (1) is equivalent to

$$\min_Y \text{RSS}(X, Y) = \|D - YX^\top\|^2 \quad \text{s.t. } Y \in \mathbb{1}^{n \times r}, \\ X = D^\top Y(Y^\top Y)^{-1}$$

The matrix Y indicates the **cluster assignments**.

The matrix X gathers the **centroids** of all clusters column-wise.

The k -means Decomposition Scheme



Example: k -means for Movie Recommender Systems

$$\begin{pmatrix} 5 & \mu & 1 & 1 \\ \mu & 1 & 5 & \mu \\ 2 & 1 & 5 & 3 \\ 4 & \mu & 4 & 2 \\ 5 & 5 & \mu & 1 \\ \mu & 1 & 5 & 3 \end{pmatrix} \approx \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 4.7 & 3.7 & 2.7 & 1.3 \\ 2.3 & 1.0 & 5.0 & 3.0 \end{pmatrix}$$

Every user's preferences are approximated by a linear combination of the rows in the second matrix:

$$(5 \quad \mu \quad 1 \quad 1) \approx 1 \cdot (4.7 \quad 3.7 \quad 2.7 \quad 1.3) + 0 \cdot (2.3 \quad 1.0 \quad 5.0 \quad 3.0)$$

Ok, so *k-means* is an instance
of the *rank- r* matrix
factorization problem.

Can we also characterize the global minimizers of k -means like we did it for the rank- r matrix factorization problem with truncated SVD?

Unfortunately not.

However, we can characterize the **global minimizers** of the objective when we **fix one of the factor matrices**.

Centroids are Part of the Solution

Theorem (Centroids as minimizers of an optimization problem)

Given $D \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{1}^{n \times r}$, the minimizer of the optimization problem

$$\min_X \|D - YX^\top\|^2 \quad \text{s.t. } X \in \mathbb{R}^{d \times r} \quad (2)$$

is given by the centroid matrix $X = D^\top Y(Y^\top Y)^{-1}$.

Proof (sketch): Show that the objective in Eq. (2) is convex. The minimizer is then given by the stationary point:

$$\begin{aligned} \nabla_X \|D - YX^\top\|^2 &= -2(D - YX^\top)^\top Y = 0 \\ \Leftrightarrow D^\top Y(Y^\top Y)^{-1} &= X \end{aligned}$$

Nearest Centroid Clusters are another Part of the Solution

Theorem (Nearest centroid clusters as minimizers)

Given $D \in \mathbb{R}^{n \times d}$ and $X \in \mathbb{R}^{d \times r}$, the minimizer of the optimization problem

$$\min_Y \|D - YX^\top\|^2 \quad \text{s.t. } Y \in \mathbb{1}^{n \times r}$$

is the matrix, assigning every point to the nearest centroid:

$$Y_{is} = \begin{cases} 1 & \text{if } s = \arg \min_{1 \leq t \leq r} \{\|X_{\cdot t} - D_{i \cdot}\|^2\} \\ 0 & \text{otherwise} \end{cases}$$

Proof (sketch): Follows from the k-means centroid objective:

$$\min_Y \sum_{s=1}^r \sum_{i=1}^n Y_{is} \|D_{i \cdot} - X_{\cdot s}^\top\|^2.$$

Lloyds' Algorithm Performs Block-Coordinate Descent

Lloyds' algorithm actually performs an alternating minimization, also called **block coordinate descent**:

$$X_{k+1} \leftarrow \arg \min_{X \in \mathbb{R}^{d \times r}} \|D - Y_k X^\top\|^2$$

$$Y_{k+1} \leftarrow \arg \min_{Y \in \mathbb{I}^{n \times r}} \|D - Y X_{k+1}^\top\|^2$$

The sequence $\{(X_k, Y_k)\}$ converges, since we decrease the objective function value in every step:

$$RSS(X_0, Y_0) > RSS(X_1, Y_1) > RSS(X_2, Y_2) > \dots \geq 0.$$

Some Notes about k -means Optimization

The k -means problem is NP-hard. (SVD is polynomially solvable!)

k -means poses a nonconvex optimization problem, and every feasible cluster indicator matrix and the corresponding centroids are one local minimum.

Hence, finding a good initialization is important! (HW).

The Most Important Slide of this Lecture

Theorem (Equivalent k -means objectives)

The following objectives are equivalent

$$\min_Y \sum_{s=1}^r \sum_{i=1}^n Y_{is} \|D_{i\cdot} - X_{\cdot s}^\top\|^2 \quad \text{s.t. } X \in \mathbb{R}^{d \times r}, Y \in \mathbb{1}^{n \times r}$$

$$\min_Y \|D - YX^\top\|^2 \quad \text{s.t. } X = D^\top Y(Y^\top Y)^{-1}, Y \in \mathbb{1}^{n \times r}$$

$$\min_{Y,X} \|D - YX^\top\|^2 \quad \text{s.t. } X \in \mathbb{R}^{d \times r}, Y \in \mathbb{1}^{n \times r}$$

The k -means algorithm
(Lloyds' algorithm) performs
block coordinate descent.