Optimization Background: Coordinate Descent

Goal: Minimize a function $J(\theta)$
- $\hat{\theta} = \arg\min_{\theta} J(\theta)$

Idea: Pick one dimension, and minimize along that dimension.

Algorithm:
1. Choose initial point $\hat{\theta}$
2. Repeat until stopping criterion is reached.
   - $\theta_1 = \arg\min J(\theta_1, \theta_2, \ldots, \theta_M)$
   - $\theta_2 = \arg\min J(\theta_1, \theta_2, \ldots, \theta_M)$
   - ...$
   - \theta_M = \arg\min J(\theta_1, \theta_2, \ldots, \theta_M)$
3. Return $\hat{\theta}$

Block Coordinate Descent

Here: An example with two blocks $\hat{x}, \hat{\beta}$, where $\hat{\theta} = [\hat{x} \hat{\beta}]$

Goal: $\hat{x}, \hat{\beta} = \arg\min J(\hat{x}, \hat{\beta})$, $\hat{x} \in \mathbb{R}^A$, $\hat{\beta} \in \mathbb{R}^B$

Idea: Minimize over an entire group of variables at a time.

Algorithm:
1. Choose initial point $\hat{x}, \hat{\beta}$
2. Repeat until stopping criterion
   - $\hat{x} = \arg\min J(\hat{x}, \hat{\beta})$
   - $\hat{\beta} = \arg\min J(\hat{x}, \hat{\beta})$
Our first example of unsupervised learning

**Goal:** partition unlabeled instances into groups of “similar” points

**Input:** Unlabeled data: \( D = \{ x^{(1)}, x^{(2)}, \ldots, x^{(n)} \} \), \( x \in \mathbb{R}^m \)

*We do not know the labels of the training examples.*

**Output:**

**View #1:**
- Labeled instances: \( \{(x^{(1)}, z^{(1)}), (x^{(2)}, z^{(2)}), \ldots, (x^{(n)}, z^{(n)})\} \)
- \( z^{(i)} \in \{1, \ldots, K\} \)
- These cluster assignments are predictions

**View #2:**
- Cluster centers: \( \{c_1, c_2, \ldots, c_K\} \)
- \( C_j = \{x^{(i)}: z^{(i)} = j\} \)
- \( C_j \) is the point in the \( j \)-th partition

**Important Questions:**
- How many clusters are there?
- How do we define “similarity” between points?

**Objective-Based Clustering**

**Example:** K-Means Objective

**Input:** \( D = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \)

**Cluster Centers:** \( C = \{c_1, c_2, \ldots, c_K\} \)

**Decision Rule:** Assign \( x^{(i)} \) to its nearest cluster center \( c_j \)

**Objective:**

\[
\bar{z} = \arg\min_{z \in \mathbb{R}^n} \sum_{i=1}^{n} \min_{j \in \{1, \ldots, K\}} \| x^{(i)} - c_j \|^2
\]

**Equivalent Objective:**

\[
\bar{z} = \arg\min_{\bar{z}} \sum_{i=1}^{n} \min_{j \in \{1, \ldots, K\}} \| x^{(i)} - c_j \|^2
\]

\[
\bar{z} = \arg\min_{\bar{z}} \sum_{i=1}^{n} \left( \sum_{j \in \{1, \ldots, K\}} \| x^{(i)} - c_j \|^2 \right)
\]

**Question:** How should we optimize \( J_{k\text{-means}}(\bar{z}, \bar{z}) \)?
Computational Complexity of K-Means Objective Minimization Problem

1. Objective is non-convex
2. NP-Hard, even for \( k = 2 \) \# clusters
   even for \( M = 2 \) \# features

**Easy Case #1: \( k = 1 \)**
\[
\hat{c}_1 = \arg\min_{c_1} \frac{1}{N} \sum_{i=1}^{N} \| x^{(i)} - c_1 \|^2
= \frac{1}{N} \sum_{i=1}^{N} x^{(i)} \text{ mean}
\]

**Easy Case #2: \( M = 1 \)**

Dynamic program in time \( O(N^2K) \)

**K-Means in Practice**
- Solve minimization problem heuristically w/ Block Coord. Descent.

**K-Means Algorithm**:

1. Given \( x^{(1)}, \ldots, x^{(n)} \)
2. Initialize cluster centers \( \hat{c} = \{ \hat{c}_1, \ldots, \hat{c}_k \} \)
   Initialize cluster assignments \( \hat{z} = \{ \hat{z}^{(1)}, \ldots, \hat{z}^{(n)} \} \)
3. Repeat until objective stops changing,
   a) \( \hat{c} = \arg\min_{c} \sum_{i=1}^{N} \| x^{(i)} - \hat{z}_i \|^2 \) \( \text{ Min over centers, w/assignments fixed} \)
   b) \( \hat{z} = \arg\min_{z} \sum_{i=1}^{N} \| x^{(i)} - c^{(i)} \|^2 \) \( \text{ Min over assignments, w/centers fixed} \)

**3) decomposes**:
\[
\sum_{i=1}^{N} \| x^{(i)} - \hat{z}_i \|^2 = \sum_{j=1}^{K} \sum_{i: x^{(i)} \in \hat{c}_j} \| x^{(i)} - c_j \|^2
\]
\[
\hat{c}_1 = \arg\min_{c_1} \sum_{i: x^{(i)} \in \hat{c}_1} \| x^{(i)} - \hat{z}_1 \|^2
\]
\[
\hat{c}_2 = \arg\min_{c_2} \sum_{i: x^{(i)} \in \hat{c}_2} \| x^{(i)} - \hat{z}_2 \|^2
\]
\[
\vdots
\]
\[
\hat{c}_k = \arg\min_{c_k} \sum_{i: x^{(i)} \in \hat{c}_k} \| x^{(i)} - \hat{z}_k \|^2
\]

Each is just Easy Case #1:
\[
\hat{c}_j = \arg\min_{c_j} \sum_{i: x^{(i)} \in \hat{c}_j} \| x^{(i)} - \hat{z}_j \|^2
= \text{ mean of points in cluster } j
= \frac{1}{N_j} \sum_{i: x^{(i)} \in \hat{z}(i)} x^{(i)}
\]

\[
\hat{z}(i) = \arg\min_{c_j} \sum_{j} \| x^{(i)} - c_j \|^2
\]

\[
\hat{z}(i) = \arg\min_{c_j} \sum_{j} \| x^{(i)} - c_j \|^2
= \text{ closest cluster center for point } x^{(i)}
\]
This is a local opt. alg. for a nonconvex objective.

K-Means just finds a local min.

⇒ How we initialize $\hat{C}, \hat{Z}$ is crucial.

**Three Options:**

1. **Random:** select points uniformly at random (w/o replacement) from dataset $D$

2. **Furthest Traversal:**
   - Select points in $D$ s.t. $c_j$ is as far as possible from $c_1, \ldots, c_{j-1}$

3. **K-Means++:**
   - Interpolate between $\text{(1)}$ and $\text{(2)}$
   - Good theoretical guarantees.