Applications

- Building blocks of ChatGPT
  - Attention mechanism
  - Cosine Similarity in embedding space (applications to Natural Language Processing)
- Clustering and k-means
  - Topic discovery
  - MNIST Digit Clustering

Topics

- Distance and Nearest Neighbors (VMLS 3.2)
- Clustering (VMLS Ch. 4)
  - Clustering objective
  - Centroids
  - k-means algorithm

Clustering

We consider the task of clustering collections of vectors into groups or clusters of vectors that are close to each other, as measured by the the distance between pairs of them.

We first define the notion of distance between two vectors, which perhaps unsurprisingly, will be closely related to the idea of norms. We then introduce the k-means algorithm, which is widely used in practical applications.

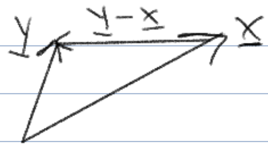
We've worked hard in the past few lectures to develop the tools needed to reason about general vectors: here we get to see some of these ideas at play!

Distance (VMLS 3.2)

We can use the Euclidean norm to define the Euclidean distance between two vectors $\underline{x}$ and $\underline{y}$ as the norm of their distance:

$$\mathrm{dist}(\underline{x},\underline{y}) = \|\underline{x}-\underline{y}\|.$$

Note that this is measuring the length of the arrow drawn from point $\underline{x}$ to point $\underline{y}$:



Although we will only work with the Euclidean distance, it can also be defined with respect to a general norm, and inherits many of the intuitive properties of Euclidean distance:

(a) Symmetry: $\mathrm{dist}(\underline{x},\underline{y}) = \mathrm{dist}(\underline{y},\underline{x})$
(b) Positivity: $\mathrm{dist}(\underline{x},\underline{y}) \geq 0$, and $\mathrm{dist}(\underline{x},\underline{y})=0$ if and only if $\underline{x}=\underline{y}$
(c) Triangle inequality: $\mathrm{dist}(\underline{x},\underline{z}) \leq \mathrm{dist}(\underline{x},\underline{y}) + \mathrm{dist}(\underline{y},\underline{z})$ for all $\underline{x},\underline{y},\underline{z}$

When the distance $\|\underline{x}-\underline{y}\|$ between vectors $\underline{x},\underline{y}\in V$ is small, we say they are "close" or "nearby." If the distance $\|\underline{x}-\underline{y}\|$ is large, we say they are "far." What constitutes "close" or "far" is typically application dependent.

Example: $\underline{x} = \begin{bmatrix}1\\2\end{bmatrix}$, $\underline{y} = \begin{bmatrix}3\\4\end{bmatrix}$, $\mathrm{dist}(\underline{v},\underline{w}) = \|\underline{x}-\underline{y}\| = \left\|\begin{bmatrix}-2\\-2\end{bmatrix}\right\| = \sqrt{2^2+2^2} = 2\sqrt{2}$

$$= \|\underline{y}-\underline{x}\|$$

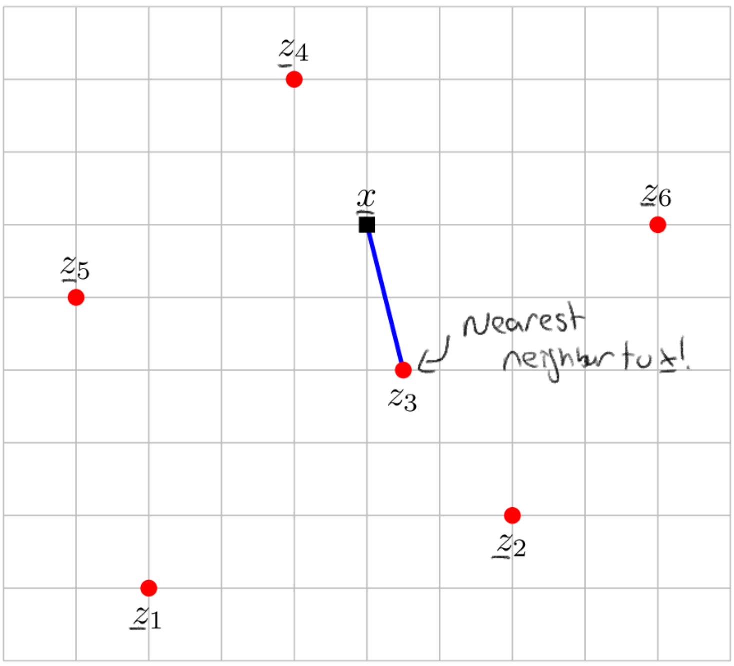
Important use cases:

- Feature distance: If $\underline{x}, \underline{y} \in \mathbb{R}^n$ are vectors containing features of two objects, $\|\underline{x}-\underline{y}\|$ is called the feature distance. It gives a measure of how "different" two objects are. For example, suppose each vector represents a patient in a hospital with entries such as age, weight, height, and test results. We can use $\|\underline{x}-\underline{y}\|$ if patients $\underline{x}$ and $\underline{y}$ are "close" to each other with respect to these features.

- Nearest neighbor: Suppose we are given a collection $\underline{z}_1, \ldots, \underline{z}_m \in V$ of m vectors living in a vector space V. We say that $\underline{z}_j$ is the nearest neighbor of $\underline{x}$ among the vectors $\underline{z}_1, \ldots, \underline{z}_m$ if

$$\|\underline{x}-\underline{z}_j\| \leq \|\underline{x}-\underline{z}_i\|, \quad \text{for } i=1,\ldots,m. \qquad (NN)$$

In words, this means $\underline{z}_j$ is the closest vector to $\underline{x}$ among $\underline{z}_1, \ldots, \underline{z}_m$. This is illustrated below; we note that the nearest neighbor may not be unique (e.g., if several $\underline{z}_i$ satisfy condition (NN)).

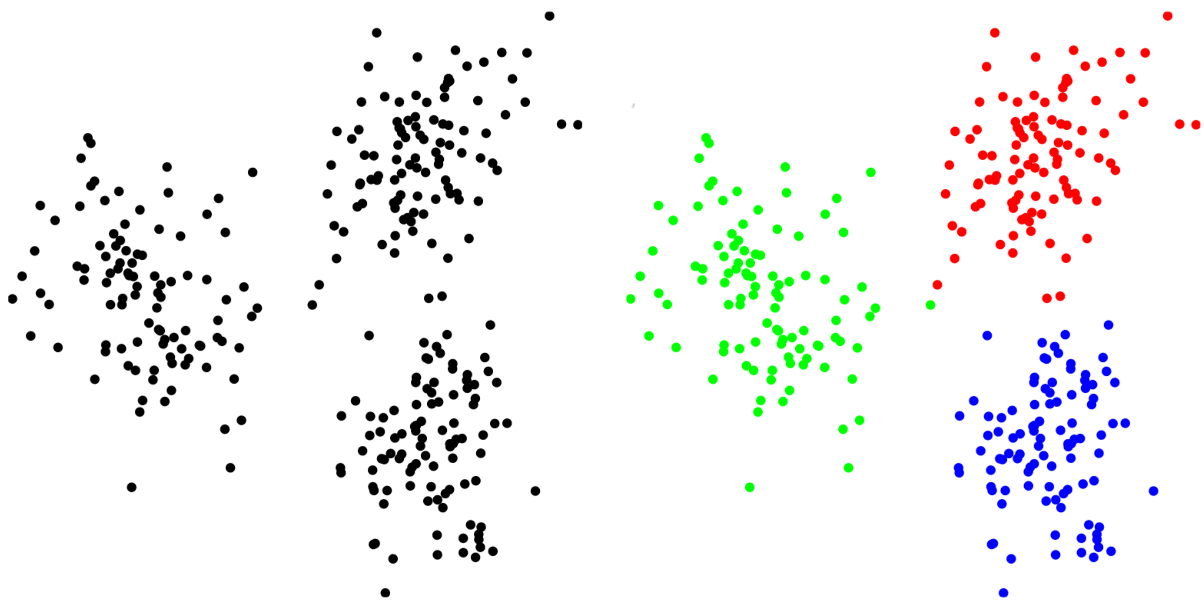


Clustering (VMLS 4.1)

Suppose we have N vectors $x_1, \dots, x_N \in V$. The goal of clustering is to group the vectors into k groups, or clusters, with the vectors in each group close to each other.

Normally the number of groups k is much smaller than the total number of vectors N. Typical values in practice for k range from a handful (2-3) to hundreds, and N ranges from hundreds to billions.

The figure below shows a simple example with $N=300$ vectors in $\mathbb{R}^2$, shown as small circles. The right picture shows what is easily seen: these vectors can be clustered into $k=3$ groups in a way that "looks right" (we will quantify this idea soon):



This example is a bit silly: for vectors in $\mathbb{R}^2$, clustering is easy. Just make a scatter plot and use your eyes. In almost all applications, vectors live in $\mathbb{R}^n$ with n much bigger than 2. Another silly aspect is how cleanly points split into clusters: real data is messy, and often many points lie between clusters. Finally, in real examples, it is not always obvious how many clusters k there are.

Despite all of this, we'll see clustering can still be incredibly useful in practice. Before we dive into more details, let's highlight a few common applications where clustering is used:

- Topic discovery: Suppose $\underline{x}_i$ are word histograms associated with N documents (a word histogram $\underline{x}$ has entries x_i which count the number of times word i appears in a document). Clustering will partition the N documents into k groups, which can be interpreted as groups of documents with the same or similar topics, genre, or author. This is sometimes called automatic topic discovery.

- Customer market segmentation: Suppose the vector $x_i \in \mathbb{R}^n$ gives the dollar values of n items purchased by customer i in the past year. A clustering algorithm groups the customers into k market segments, which are groups of customers with similar purchasing patterns

Other examples include patient, zip code, student, and survey response clustering, as well as identifying weather zones, daily energy use patterns, and financial sectors. See pp. 70-71 of VMLS for more details.

A Clustering Objective (VMLS 4.2)

Our goal now is to formalize the ideas described above, and introduce a quantitative measure of "how good a clustering" is.

Specifying cluster assignments:

We specify a clustering of vectors by assigning each vector to a group. We label the groups $1, \ldots, k$, and assign each of the N vectors $\underline{x}_1, \ldots, \underline{x}_N$ to a group via the vector $\underline{c} \in \mathbb{R}^N$, with c_i = group # that x_i has been assigned to. For example, if $N=5$ and $k=3$, then

$$c = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}$$

assigns $\underline{x}_1$ to group 3, $\underline{x}_2, \underline{x}_3, \underline{x}_4$ to group 1, $\underline{x}_5$ to group 2.

We will also describe clusters by the sets of indices for each group, with G_j the set of indices associated with group j. For our simple example, we have

$$G_1 = \{2,3,4\}, \quad G_2 = \{5\}, \quad G_3 = \{1\}$$

In general, we have that $G_j = \{ i \mid c_i = j \}$.

Group Representatives:

Each group is assigned a group representative $z_1, \ldots, z_k \in V$. Note that these representatives can be any vector, and need not be one of the given vectors $x_1, \ldots, x_N$. A good clustering will have each representative close to vectors in its associated group, i.e.,

$$\text{dist}(\underline{x}_i, \underline{z}_{c_i}) = \|\underline{x}_i - \underline{z}_{c_i}\|$$

is small for all $i = 1, \ldots, N$. Note that according to our notation, $\underline{x}_i$ is in group C_i, so $\underline{z}_{c_i}$ is the group representative against which $\underline{x}_i$ should be measured.

A Clustering Objective:

We now define a clustering objective that assigns a score, or cost, to a choice of clustering and representatives:

$$J_{clust} = \left(\|\underline{x}_1 - \underline{z}_{c_1}\|^2 + \|\underline{x}_2 - \underline{z}_{c_2}\|^2 + \cdots + \|\underline{x}_N - \underline{z}_{c_N}\|^2\right)/N$$

which computes the mean square distance from the vectors to their associated representatives. The smaller J_{clust} is, the "better" the clustering. (What does it mean if $J_{clust} = 0$?).

A clustering is said to be optimal if the choice of group assignments $c_1, \ldots, c_N$ and group representatives $\underline{z}_1, \ldots, \underline{z}_k$ lead to the smallest achievable clustering objective J_{clust} - in that case, these choices are said to minimize the objective J_{clust}. Unfortunately, except for very small problems, it is computationally prohibitive to find an optimal clustering.

Fortunately, the k-means algorithm we will introduce next can be run effectively on very large problems, and often finds very good clusterings that achieve objective values J_{clust} near the smallest possible value. Because k-means finds such suboptimal solutions, we call it a heuristic. Heuristics are often looked down on in more theory oriented circles because they cannot guarantee the quality of their solutions, but as we'll see, they often work incredibly well in practice.

The idea behind k-means is to break down the overall hard problem of choosing the best representatives and clusterings at the same time into two subproblems we can easily solve effectively. We step through these next.

Partitioning Vectors with Representatives Fixed

Pretend for a moment that we have already found group representatives $z_1, \dots, z_k$, and our task is to pick the group assignments $c_1, \dots, c_N$ which lead to the smallest possible J_{clust}. This problem can be solved easily using the idea of nearest neighbors we saw earlier.

Notice that the objective J_{clust} is a sum of N terms, with one term for each vector x_i. Further, the choice of c_i (i.e., the group to which we assign x_i) only affects the term

$$\frac{1}{N} \|x_i - z_{c_i}\|^2$$

in J_{clust}, so we can choose c_i to make this term smallest since it doesn't affect any other terms in our clustering objective.

To minimize $\|x_i - z_{c_i}\|^2$, we simply pick the c_i that makes this as small as possible, i.e., pick the c_i so that

$$\|x_i - z_{c_i}\| \leq \|x_i - z_j\| \text{ for } j = 1, \dots, k.$$

This should look familiar! Modulo our new notation, we should assign x_i to its nearest neighbor among the representatives.

Optimizing the Group Representatives with Assignments Fixed:

Now we flip things around, and assume each vector x_i has been assigned to a group c_i. How should we pick group representatives $z_1, \dots, z_k$ to minimize J_{clust}? We start by rearranging our objectives into k sums, one for each group.

$$J_{clust} = J_1 + J_2 + \cdots + J_k$$

where $J_j = \frac{1}{N} \sum_{i \in G_j} \|x_i - z_j\|^2$ is the contribution to J_{clust} from the vectors in group j. The sum notation here means we should include terms $\|x_i - z_j\|^2$ in our sum if $i \in G_j$ (i.e., if x_i has been assigned to group j).

The choice of representative z_j only affects the term J_j, so we can choose z_j to minimize J_j. You can check, e.g., using vector calculus, that the best choice is to pick z_j to be the average (or centroid) of the vectors in group j:

$$z_j = \frac{1}{|G_j|} \sum_{i \in G_j} x_i$$

Here $|G_j|$ is the cardinality of the set G_j, and denotes the number of elements in the set G_j, i.e., the size of group j.

The k-Means Algorithm

While we can't yet solve the problem of jointly choosing group assignments & group representatives to minimize J_{clust}, we know how to solve for one component when the other is fixed.

The k-means algorithm produces an approximate solution to the clustering problem by iterating between the two subroutines. A key feature of this approach is that J_{clust} gets better or stays the same with each iteration, meaning it is guaranteed to converge, possibly (likely) to a suboptimal solution.

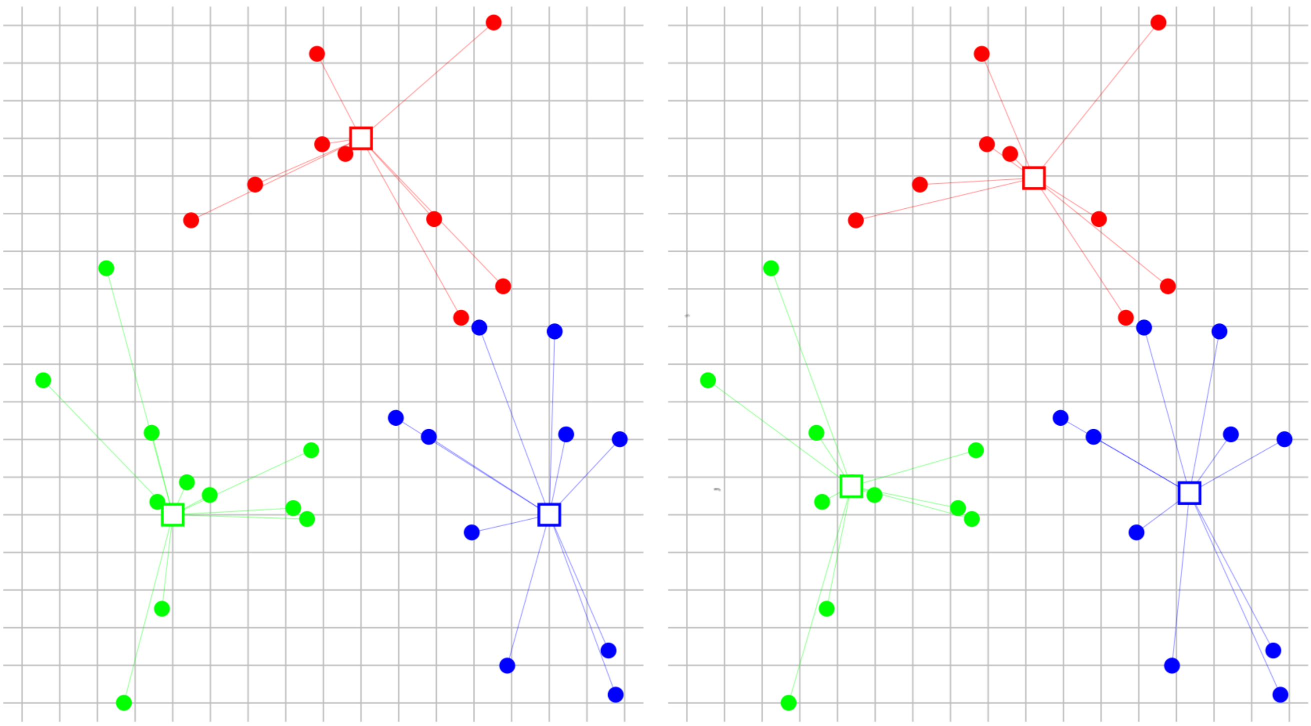
Algorithm 4.1 k-MEANS ALGORITHM

given a list of N vectors $\underline{x}_1, \ldots, \underline{x}_N$, and an initial list of k group representative vectors $\underline{z}_1, \ldots, \underline{z}_k$

repeat until convergence

1. *Partition the vectors into k groups.* For each vector $i = 1, \ldots, N$, assign x_i to the group associated with the nearest representative.
2. *Update representatives.* For each group $j = 1, \ldots, k$, set $\underline{z}_j$ to be the mean of the vectors in group j.

One iteration of the k-means algorithm is illustrated below:

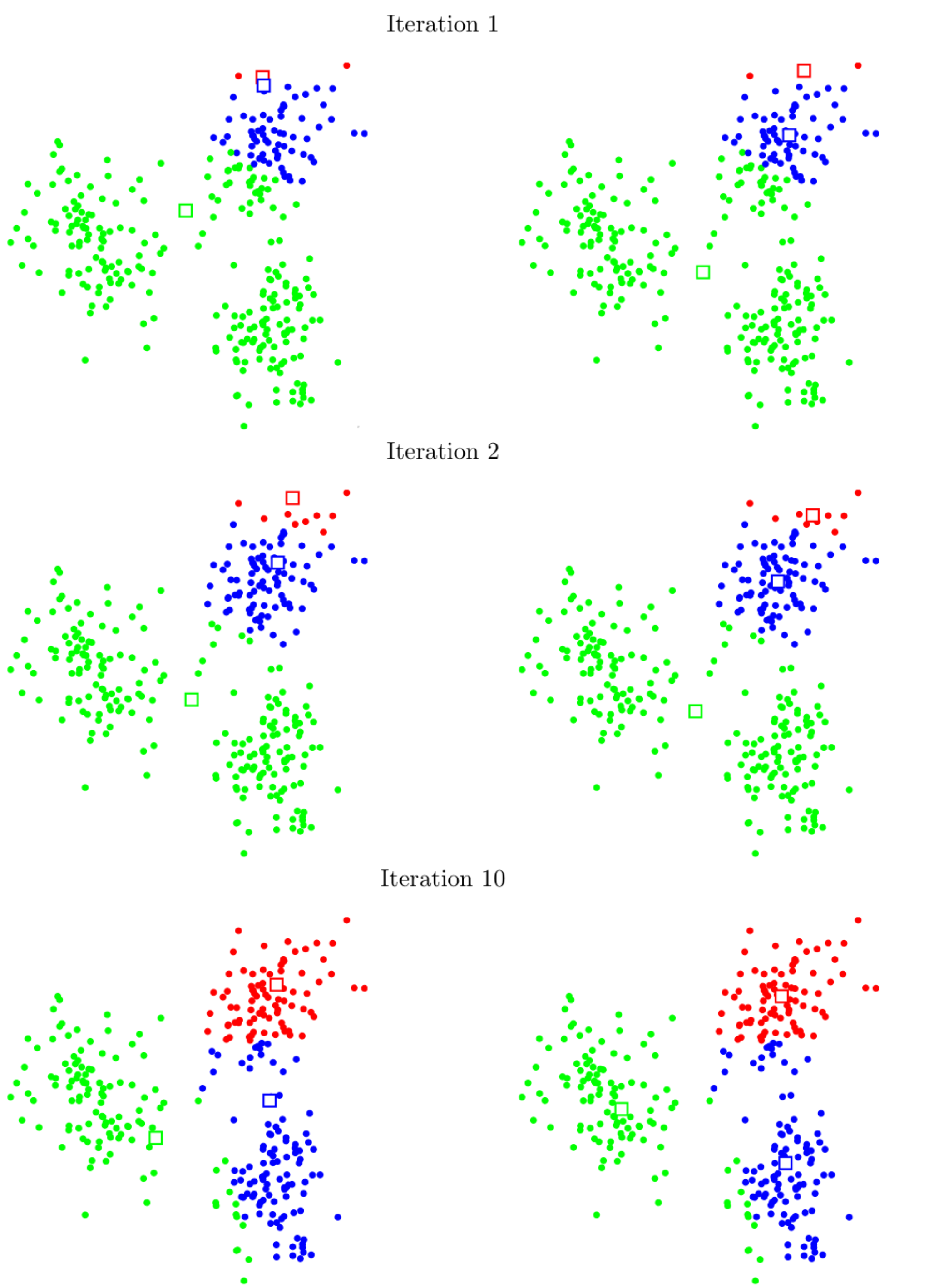


Left: vectors $\underline{x}_1, \ldots, \underline{x}_N$ are assigned to the nearest representative $\underline{z}_j$.
Right: the representatives are updated to the centroids of the new groups.

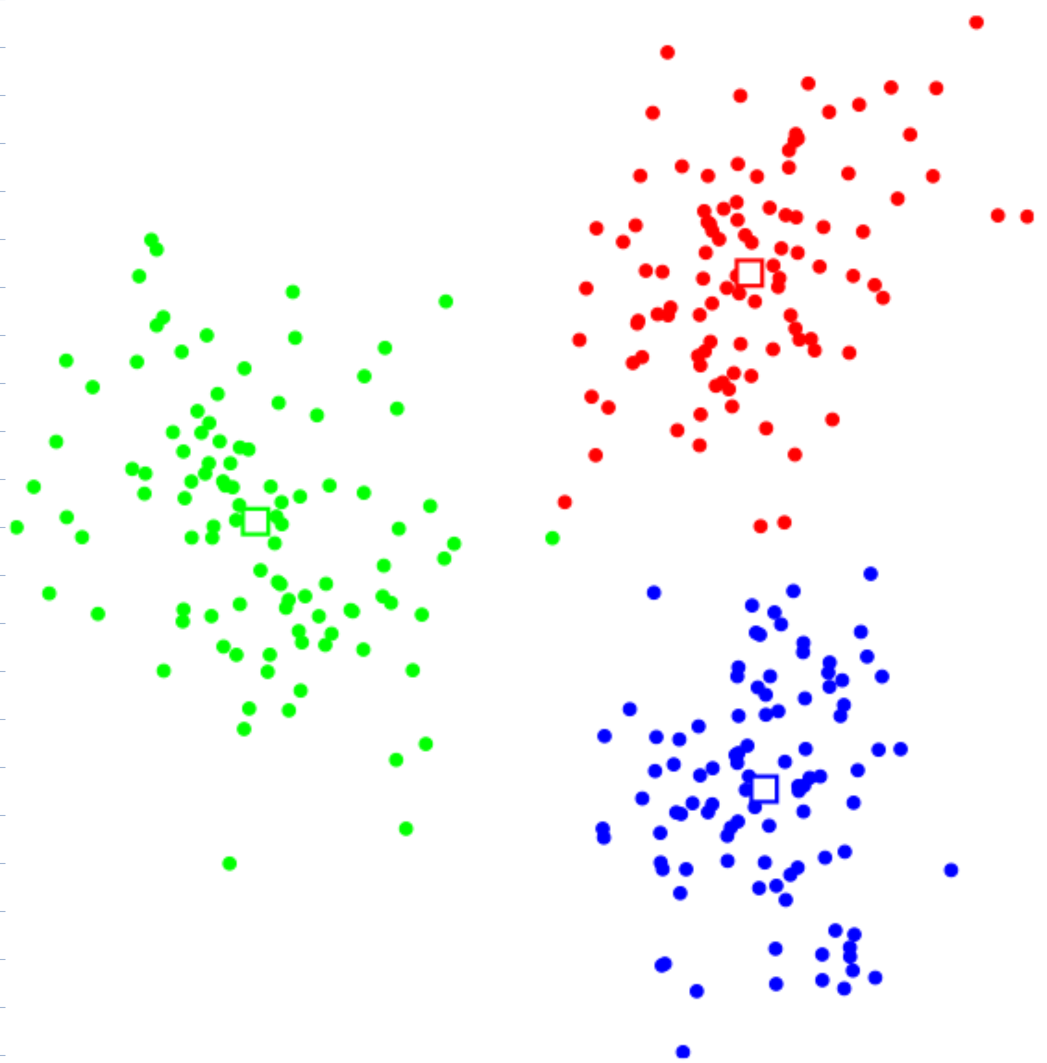
Some comments and clarifications:

- Ties can be broken by assigning x_i to the tied group of smallest j (or any other deterministic rule — assigning at random can affect convergence of the algorithm).
- If at any time a group is empty, it (and its representative) are simply dropped.
- If group assignments don't change during an iteration, then the representatives will also stay the same: this is what we mean by the algorithm converging.
- There are many ways to initialize the group representatives: a common approach is to pick them at random from the x_i.

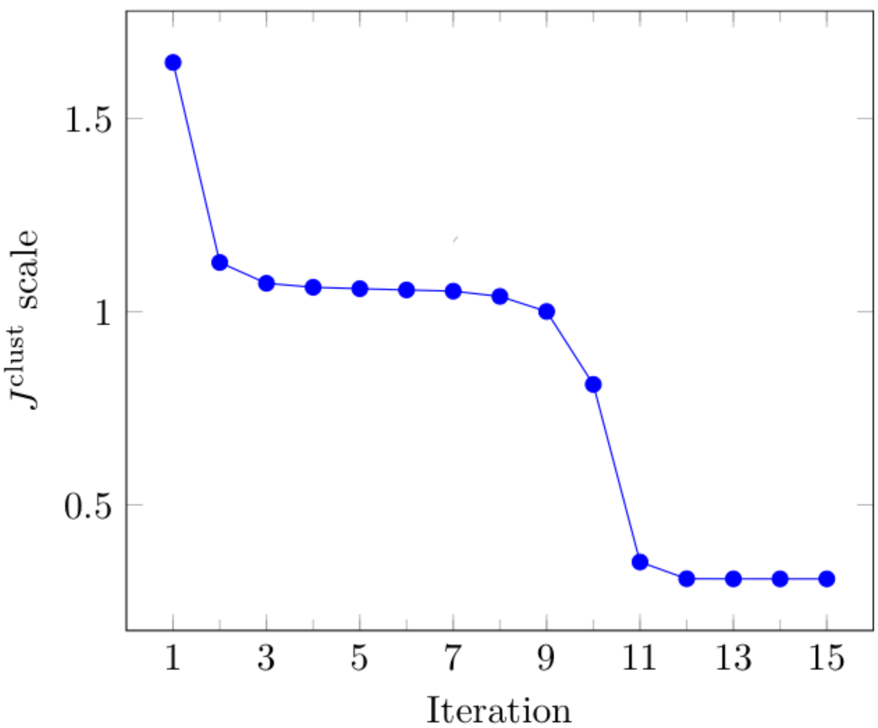
Toy example:



Final Clustering:



J_{clust} Convergence:



In the online notes, you will see a more interesting example as applied to topic discovery (also see pp 82-85 of UMLS).

In the homework, you will implement and test your own version of k-means and use it for handwritten digit recognition/classification.