Grouping the dataset based on the similarity

http://en.wikipedia.org/wiki/K-medoids

K –medoids

Entry for this application will be the file with all points that was generated in the first laboratory (called the "points file"). This file will be considerate the training dataset for this learning method. This algorithm is an unsupervised learning algorithm because in the dataset are specified only the input data without specifying the category (for us the center or color) to which they belong.

The algorithm steps:

1. It is randomly chooses a number of "centers of classes" depending on how many categories we want to get at the end. It is recommended to be chosen a randomly number between 2 and 10.
2. For each center we will select randomly a point from the "points file" and we will initialize the center coordinates with the point coordinates. Thus, the coordinates will be in the same space of the training data. This step is the step where we randomly set the possible centers of classes. For each generated class will be assigned a different color.
3. We will take a point from the "point file" and we will compute the similarity between it and all the centers generated in the step 2. The point would be assigned to the class which is closest (similar). In other words, the point will be classified at the nearest neighbor. There are several methods for computing the similarity. In the next paragraph we present 2 such methods. You can suggest other methods.
4. Repeat the step 3 for all the points from the "points file".
5. After classifying all of points from the "points file" (after assigned a class to each point from the file), we will calculate the center of gravity of all items (points) classified into a class. Thus for each class will compute a new center of gravity. All items will be colored with the class color to which they were assigned for a good view of the algorithm operation.
6. Find the point that is the nearest point of the center of gravity computed in the previous step and modify the class center coordinate with the point coordinate (move the center of classes in the point position).
7. Resume the algorithm from the step 3 as long as there are a major difference between the position of the old center of a class and the position of the new center of a class. The classifications made in step 3 will be deleted. If is not a major differences between the old and the new class position go to step 8.
8. All points are drawn on the screen, with different color, according to the gangway where they belong for a better visual analysis about the algorithm operation.

Note:

1. For a good view of the algorithm operation it is recommended to show on screen the points colored corresponding after each execution step.
2. The center of gravity of a class may be, for example, the arithmetic mean of all elements that are contained in that class.

Methods for compute the similarity

There are a lot of methods for compute the similarity. Each method is selected according to the applicability domain. Among of the most common methods are: computing the Euclidean distance or the cosine angle between two vectors. Because those methods compute the distance between two vectors these methods must respect all the distance proprieties. I present in this laboratory two methods Euclidean distance and Manhattan distance.

1. Compute the similarity using the Euclidean distance:

$d(\vec{x},\vec{x}') = \sqrt{\sum_{i=0}^{n}(x_i - x'_i)^2}$, where *n* represent the number of features, *x* and *x*' represents the input vectors (items, points).

2. Compute the similarity using the Manhattan distance:

$d(\vec{x},\vec{x}') = \sum_{i=0}^{n}(|x_i - x'_i|)$