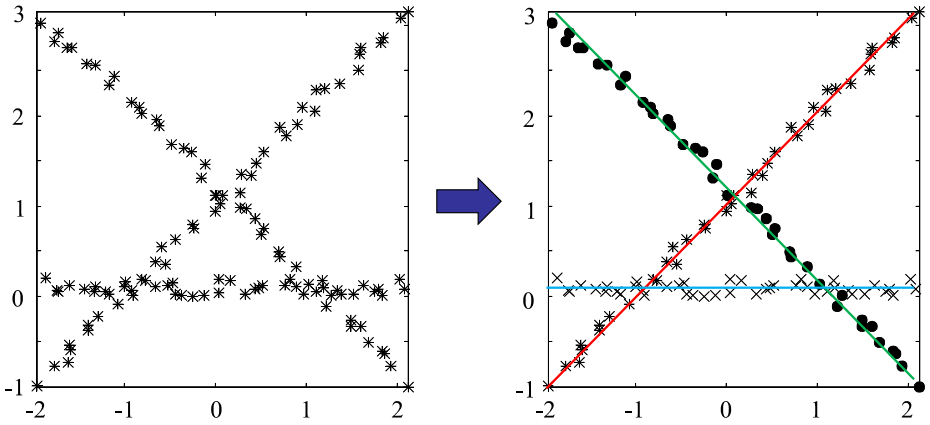
Computational Intelligence and Applications / NCTU Spring 2017
Program Assignment #2 (due: 4/26/2017)

You can use any programming language and platform of your choice: C++, C#, Matlab, Java, python, ... I will not actually run your code.

The goal of this program assignment is to implement an evolutionary algorithm to solve a problem. Our target problem here is to cluster a set of 2-D points into straight lines; each point is assigned to one of the lines. This is illustrated below:



Note: While there are more efficient methods for this task, you are NOT allowed to use those.

There are two ways to code the clustering result (i.e., two different representations of genotypes). You can use either one, or better yet, you can try both in order to compare them. Here let there be N points and we want M lines, with both N and M assumed to be known.

Representation #1: The genotype consists of the cluster assignments of the points. Each individual has N genes, each can be an integer in the range of $1 \ldots M$.

Representation #2: The genotype consists of the parameters of the M lines. Each line is parameterized in the form of $x\cos\theta + y\sin\theta = \rho$. Here ρ is the distance between the line and the origin, and θ is the angle from the y axis to the direction of the line. The typical approach is to limit θ to $0 \sim \pi$ and allow ρ to take positive or negative values. Alternatively, you can allow $0 \le \theta < 2\pi$ and require that $\rho \ge 0$. In total, there are $2M$ real-valued genes.

Fitness: The fitness is given by the mean squared distance of all the points to their assigned lines. To be able to do this, you need to be able to compute the distances between points and lines.

> For representation #1, you can determine a line from its points in the following way: Let $\boldsymbol{x}_1 \ldots \boldsymbol{x}_n$ be column vectors for the n points belonging to the line. Let $\boldsymbol{\mu}$ be their mean vector. The line passes through $\boldsymbol{\mu}$ and its direction is along the eigenvector with the largest eigenvalue of the covariance matrix of $\boldsymbol{x}_1 \ldots \boldsymbol{x}_n$.
>
> For representation #2, you have to first assign each point to one of the lines. The rule is to assign a point to its closest line.

The other components (initialization, control flow, crossover, mutation, selection, termination) of EA are left for you to experiment with. Some of them are representation dependent and some are not. I will provide a few test datasets for you to play with.

You need to submit a report (limited to 10 pages) describing

- Methods you have implemented.
- Experiments you have done, and the results.

- Analysis - Are the results what you expect? Why?

Include your code listing as an appendix of your report. The code should be well documented. The code listing is not included in the 10-page limit.

Submit your report electronically through e3.

The grading is based on the following:

- Correctness of your implementation
- Quality of your experiments and analysis
- Quality of your presentation
- Quality of your code and documentation

Late submission policy: 5% credit deduction for each day late; up to 7 days late accepted.

Note: You are only required to submit 3 of the 4 programming assignments that will be posted. So you can choose not to do this one, which means that you will need to do all the other three.