Vector Field K-Means Code

October 20, 2013

1 Compiling the Code

The code has no dependencies, therefore compiling it is very simple. A makefile is included. In a system in which *make* is available just go to the src directory in the repository and type make in your terminal. This generates an executable called *vfkm*. Running this executable with no parameters shows the proper usage as in the following example

```
./vfkm
./vfkm trajFile gridRes numVecFields smoothnessWeight outDir
```

This shows that the executable receives 5 parameters, described in the following

1. The *trajFile* parameters is the path to the file containing the trajectory data to be clustered. this file should be formatted as in section 3.

2. The *gridRes* parameter is an integer number corresponding to the resolution of the grid used to estimate the vector fields.

3. The *numVecFields* parameter is the number of vector fields (i.e. cluster) to be estimated (computed).

4. The *smoothnessWeight* parameter corresponds to a real value between 0 and 1 that corresponds to how important the smoothness of the vector fields is to the optimization (see the Vector Field K-Means paper for more details).

5. The *outDir* parameter is the path of the directory in which the results of the algorithm are going to be placed. See section 5 for a description of the algorithm output.

2 Running Example

To illustrate the usage of the code as well as the input and output files we are going to use a concrete example. Assume that we have a data set composed
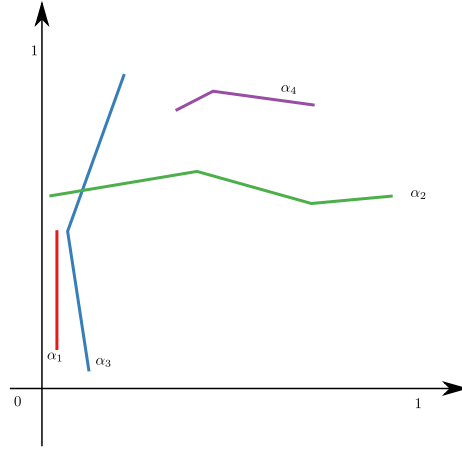
Figure 1: Trajectories in the example.

of 4 trajectories that we want to cluster. These trajectories are given by the following sequence of space-time points (i.e., the two first coodinates correspond to space coordinates and the third coordinate corresponds to time)

- $\alpha_1 = (0.1, 0.2, 0), (0.1, 0.5, 1)$

- $\alpha_2 = (0.05, 0.6, 0), (0.45, 0.65, 0.25), (0.75, 0.55, 0.5), (0.95, 0.6, 1)$

- $\alpha_3 = (0.2, 0.1, 0), (0.15, 0.5, 0.5), (0.25, 0.95, 1)$

- $\alpha_4 = (0.4, 0.8, 0), (0.5, 0.85, 0.5), (0.75, 0.825, 1)$

These trajectories are shown in figure 1.

3   Data Input

In this section we describe the format of the trajectory file used by the code. The first line of this file contains the bouding box of the region used to estimate the vector fields (Ω in the paper notation). This line follows the format "left right bottom top startTime endTime". After this first line contains the trajectory data. Each line contains the space time coordinates of the trajectories and distinct trajectories are separated by a line containing "0.0 0.0 0.0". For example, let's assume that we want to compose the file containing the trajectories of the previous section and also that we want to estimate the vector fields in the square $[0, 1] \times [0, 1]$ and the time bounds are also $[0, 1]$. The result file is the following

```
0  1  0  1  0  1
0.1  0.2  0
0.1  0.5  1
```

```
0.0 0.0 0.0
0.05 0.6 0
0.45 0.65 0.25
0.75 0.55 0.5
0.95 0.6 1
0.0 0.0 0.0
0.2 0.1 0
0.15 0.5 0.5
0.25 0.95 1
0.0 0.0 0.0
0.4 0.8 0
0.5 0.85 0.5
0.75 0.825 1
0.0 0.0 0.0
```

Let's assume that this file is saved as *trajectories.txt*. Figure 2 highlights the different portions of the file format.
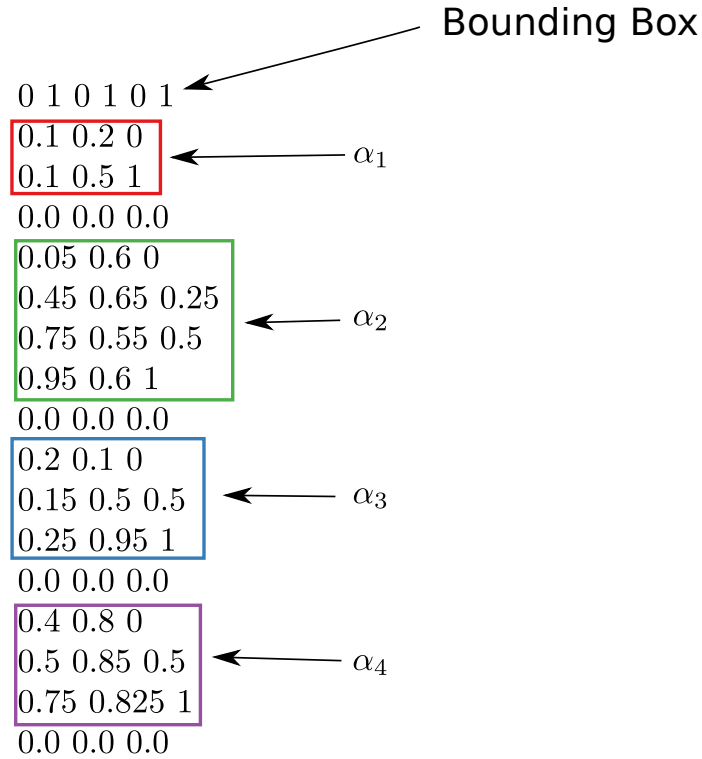

Figure 2: The *trajectories.txt* file example.
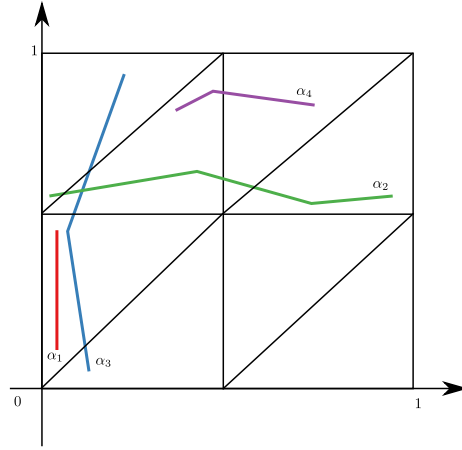
Trajectories not contained in this grid are tesselated.

Figure 3: Grid example.

4 Running the Algorithm

Now we are ready to run the algorithm. Assume that we want to find 2 clusters (estimate 2 vector fields), using a grid of resolution 3 (i.e., the given bounding box is going to be splitted using a triangular grid 18 triangles as in figure 3), and a smoothness weight of 0.5. Furthermore, we want the output to be placed in the directory called *finalOutput*. We do this by

```
./vfkm trajectories.txt 3 2 0.5 finalOutput
Loading Files...
Loading data
CALLING THIS
Optimizing...
Before optimization: 1e+20
After optimization: 0.320979
After assignment: 0.320979 changes: 0
Cleaning Memory
```

The output files should now be in the directory *finalOutput*. Let's check!

```
ls finalOutput
curves_r_0.txt   curves_r.txt     vf_r_0.txt   vf_r.txt
curves_r_1.txt   experiment.txt   vf_r_1.txt
```

In the following section we describe each of these files.

5 Algorithm Output

The algorithm generates a series of files in the output directory as shown previously. You should only worry about the *curves_r_i.txt* and *vf_r_i.txt* files, since these contains the assingment of curves to clusters and the vector fields. For each $i = 0, 1, ..., numVecFields$ (the number of clusters passed as a parameter when the code was run), these pair of files is going to be created.

5.1   Curve Files

The curve files are in the following the format

```
curveIndex error
curveIndex error
...
curveIndex error
```

where *curveIndex* corresponds to the index of the curve in the input file (the order in which the curves were given) and error corresponds to the residual in the optimization problem corresponding to that curve.

As an example, let's look at the curve files generated in the last example. The *curves_r_0.txt* file is

```
0  0.00389544
2  0.0980094
3  0.0244096
```

This indicates that the curves α_1, α_2, and α_4 (indices 0,2, and 3) were put in the cluster 0. The file *curves_r_1.txt* is

```
1  0.141112
```

which indicates that the curve α_2 was put by itself in cluster 1.

5.2   Vector Field Files

The vector field files contain the estimated vector field. As described in the Vector Field K-Means papers, vector fields are given by their values in the vertices of the grid. The vector field files are in the following format

```
numVertices
xComponent yComponent
xComponent yComponent
...
xComponent yComponent
```

The first line contains the number n of vertices that are used to define this vector field. The following n lines contain the values of the vector fields (first the x component and after that the y component) for each of the vertices starting from the bottom left vertex in the grid ending in the top right one (going each row at a time from left to right). Figure 4 illustrates the ordering of the vertices in the vector field files.
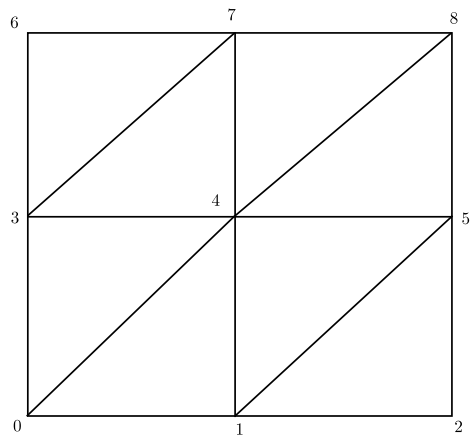
Figure 4: Vertex ordering.