Clustering Analysis of Mobility Data

Miriam Wagner, Martin Breuer, Moritz Werthebach, Timo Bergerbusch, and Walter Schikowski

RWTH Aachen, Templergraben 55, 52062 Aachen, Germany

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: Clustering · Rapidminer · Cluster · Data Mining

1 Introduction

2 Preprocessing

In order to classify the given data into smaller test sets or mask different aspects, we have to perform analysis.
We observe that even though we have 124979 individual lines defining a movement, there is one line defining a `NotANumber`-exception and therefore gets neglected for further usage.
We provide the `testDataGenerator` python script. Through flags and input arguments the script is able to create all test sets considered by our clustering and neural net approaches.
We observe the following distribution over the whole dataset:

strata	1	2	3	4	5	6	Σ
abs	6963	52265	49404	8772	5536	2038	124978
%	5.57	41.82	39.53	7.02	4.43	1.63	100

We observe that there is an upper bound on equal distribution through strata 6. It has at most 2038 individual elements.

In addition to the original paper we compute the value `ID`, which is used to combine movements considered to be from the same person. We consider two movements to coincide on the underlying person, if and only if they are consecutive in the original dataset and have the same strata, age and gender.

strata	1	2	3	4	5	6	Σ
abs	3153	23367	21418	3497	2083	595	54113
%	5.83	43.18	39.58	6.46	3.85	1.1	100

So we also have through strata 6 an upper bound of 595 for equally distributed person vector data (see Section 2.1).

2.1 Vector

As stated before, instead of simple IDs for every person we expand the parsing by using a data encapsulating in a class called `Person`. This class stores the ID,

the parameters defining a person , and all movements from that person.
Then we are able to compute the following vector, with 848 entries, for further usage, that combines all movements of the person:

$$\underbrace{\#o_1, \dots, \#o_{413}, \#d_1, \dots, \#d_{413}}_{2\cdot 413}, \underbrace{AM, MD, PM, MN}_{4}, \underbrace{\#r_1, \dots, \#r_7}_{7},$$
$$\underbrace{\#MoT_1, \dots, \#MoT_7}_{7}, \underbrace{SDest, SDist, G, A, strata, strataGrouped}_{6}$$

with the following abbreviations ($1 \le i \le 413$, $1 \le j \le 7$):

o_i:	the i-th origin data point	MoT_j:	the j-th mean of transportation
d_i:	the i-th destination data point	$SDest$:	sum of all durations
AM:	movements at time stamp AM	$SDist$:	sum of all distances
MD:	movements at time stamp MD	G:	the gender
PM:	movements at time stamp PM	A:	the age
MN:	movements at time stamp MN	$strata$:	the strata (used for comparison)
r_j:	the j-th reason	$strataGrouped$:	the aggregated stratas

3 Predicting

3.1 Classification

3.2 Neural Net

For all the neural net computations done we considered person vector data sets of different sizes (c.f. Section 2.1).
We do this, because results on the normal datasets had an unacceptable performance. An example is given in Figure 1.

accuracy: 59.76% +/- 2.20% (mikro: 59.76%)

	true s_1	true s_2	true s_5	true s_4	true s_3	true s_6	class precision
pred. s_1	933	635	33	28	248	5	49.57%
pred. s_2	3417	31605	288	765	10603	70	67.61%
pred. s_5	67	291	2566	852	552	138	57.46%
pred. s_4	204	1039	774	2983	2392	253	39.02%
pred. s_3	2335	18617	1533	3866	35315	292	57.00%
pred. s_6	7	78	342	278	294	1280	56.16%
class recall	13.40%	60.47%	46.35%	34.01%	71.48%	62.81%	

Fig. 1: An example of a neural net trained without person vector data.

In the following we consider 3 neural nets $\mathcal{N}_1, \mathcal{N}_2$ and $\mathcal{N}_3$, all having 4 hidden layers, 50 epochs and 10 iterations. As an example of other strata aggregation we combine the stratas 1–2, 3–4 and 5–6 together and call them $\mathcal{N}_i^\star$, for $i \in \{5, 10, 20\}$. This builds a superset of the original stratas and since the stratas themselves are logically connected this task should be easier to fulfill. The sets are provided by the `testDataGenerator` from Section 2.

	#		Set size		
Name	Neurons	AG	100	200	595
$\mathcal{N}_5$	5	✗	60.03	59.92	60.18
$\mathcal{N}_5^\star$	5	✓	87.6	89.7	71.05
$\mathcal{N}_{10}$	10	✗	75.83	73.54	69.56
$\mathcal{N}_{10}^\star$	10	✓	92.93	93.48	74.58
$\mathcal{N}_{20}$	20	✗	75.45	71.14	61.87
$\mathcal{N}_{20}^\star$	20	✓	92.87	94.4	78.32

Fig. 2: The accuracy values of the neural nets. (See excel-spreadsheet)

For each we performed 5 independent runs and take an average over those accuracy values in order to have a sophisticated statement.

The size of larger nets in terms of neurons is counter-productive, since if we take 50 neurons per layer we have $14 \cdot 50^4 \cdot 6 \approxeq 525.000.000$ synapses for which the input data set would be to small to have sufficient training.