HEURISTIC ALGORITHMS - COURSE PROJECT

Luca Cappelletti
Prof. Roberto Cordone

6 CFU

2019
IT Master Degree
University of the Studies of Milan
Italy
19 marzo 2019

Indice

1 Project Goal

1.1 The problem

The problem that will be the main focus of this project is to merge two **uniform datasets**, without duplicating content.

Definition 1.1.1 (Uniform dataset). An uniform dataset U_{D_1} is a indexed table whose rows and columns are supposed to being extracted from a discrete distribution D_1.

	Column 1	…	Column n
Index 1	4	…	3
⋮	⋮	…	⋮
Index m	9	…	6

Tabella 1.1: Uniform dataset example

Definition 1.1.2 (Column alignment). A column alignment is a procedure to align the columns of two datasets.

Definition 1.1.3 (Row alignment). A row alignment is a procedure to align the rows of two datasets.

Definition 1.1.4 (Full alignment). A full alignment is a procedure to align the columns and rows of two datasets.

1.1.1 Example

Let's suppose we have a full set of columns $C = \{C_1, C_2, C_3\}$ and a full set of rows $R = \{R_1, R_2, R_3\}$. We consider the following two datasets, extracted uniformly from a distribution D, and we want to merge them as follows.

	C_1	C_2
R_1	4	0.3
R_3	9	0.6

Tabella 1.2: First dataset, U_{1_D}

	C_1	C_3
R_2	9	"$ciao$"
R_3	7	"$johnny$"

Tabella 1.3: Second dataset, U_{2_D}

	C_1	C_2	C_3
R_1	4	0.3	
R_2	9		"$ciao$"
R_3	8	0.6	"$johnny$"

Tabella 1.4: Merged datasets, $U_{1_D} \cup U_{2_D}$

1.1.2 Project goal

The final goal is to **fully align** any number of uniform datasets from the same distribution, but since aligning two rows is relatively simple (by using weighted MSE) once the columns are aligned, the main focus of this project will be to create a procedure of **Column alignment** that reliably aligns columns.

2

Column alignment procedure

2.1 Bayesian optimization

Definition 2.1.1 (Bayesian optimization). The process of Bayesian optimization is a sequential design strategy for **global optimization** of **black-box functions** that doesn't require derivatives.

Since the objective function is unknown, the Bayesian strategy is to treat it as a random function and place a **prior** over it. The prior captures our beliefs about the behaviour of the function. After gathering the function evaluations, which are treated as data, the prior is updated to form the posterior distribution over the objective function. The posterior distribution, in turn, is used to construct an acquisition function (often also referred to as infill sampling criteria) that determines what the next query point should be.

Definition 2.1.2 (Known negatives). The known negatives are the columns from the same dataset, that therefore are assumed as not alignable any further.

The bayesian optimization determines a number of hyperparameters:

Weights vector $\underline{\omega}$: stochastic vector used for combine convexly the n metrics matrices.

Percentages vector $\underline{\rho}$: percentage of *known negatives* that can be considered acceptable.

2.1.1 Convergence

The algorithm reaches convergence when either no further significant enhancements seem to be achievable on the score or a maximum number of iterations is reached.

2.2 Column alignment sub-routine

We use the *known negative rows* to determine for each dataset a minimum weighted MSE distance ψ: when choosing to align multiple rows, we use the minimal distance for the various dataset from which the rows are extracted.

The sub routine determines:

1. For each metric i and afterwards for the weighted matrix $\boldsymbol{W}$:
   (a) An activation threshold is determined to allow at most a percentage ρ_i of known negative to be even allowed into the next step: all the elements above the threshold are given a cost $+\infty$.
   (b) The Hungarian algorithm is then run on the matrix $\boldsymbol{M}_i$ and the maximal assignment problem is solved.
   (c) The strongly connected components, determined using the Tarjan algorithm.
   (d) The aligned rows are determined using the weighted MSE.
2. The weighted matrix is determined: $\boldsymbol{W} = \sum_{i=1}^{n} \omega_i \boldsymbol{M}_i$. The same procedure as the single metrics matrices is run on $\boldsymbol{W}$.
3. The iteration Bayesian prior is obtained.

2.2.1 Bayesian prior

Definition 2.2.1 (Used Bayesian prior). We determine Bayesian optimization iteration score using the following loss formula:

$$-\sum_{i=1}^{m} \frac{\#\{\hat{y}_{\text{rows}}\}}{\max\{1, \#\{y_{i_{\text{rows}}}\}\}} \text{loss}\left(y_{i_{\text{rows}}}, \hat{y}_{\text{rows}}\right) \cdot \text{loss}\left(y_{i_{\text{cols}}}, \hat{y}_{\text{cols}}\right)$$

Figura 2.1: Prior

$y_{i_{\textbf{rows}}}$**:** the rows predicted by the i-th metric matrix.

$\hat{y}_{\textbf{rows}}$**:** the rows predicted by the weighted metrics matrix.

$y_{i_{\textbf{cols}}}$**:** the columns predicted by the i-th metric matrix.

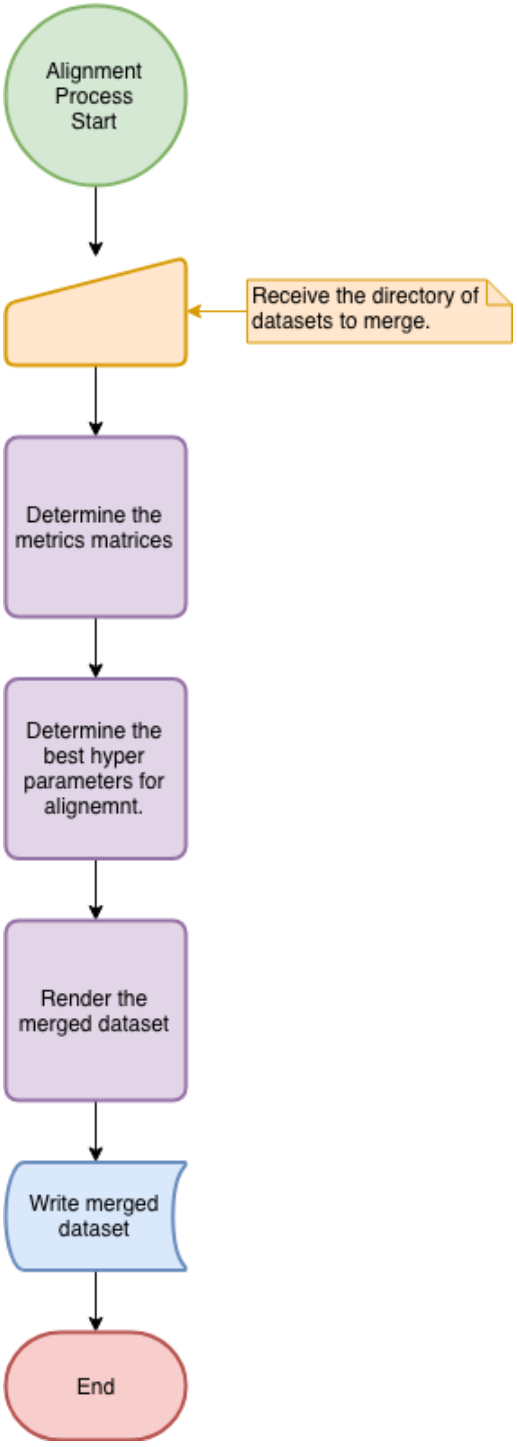
$\hat{y}_{\textbf{cols}}$**:** the columns predicted by the weighted metric matrix.

$\#\{y_{i_{\textbf{rows}}}\}$**:** number of rows aligned by the metric matrix.

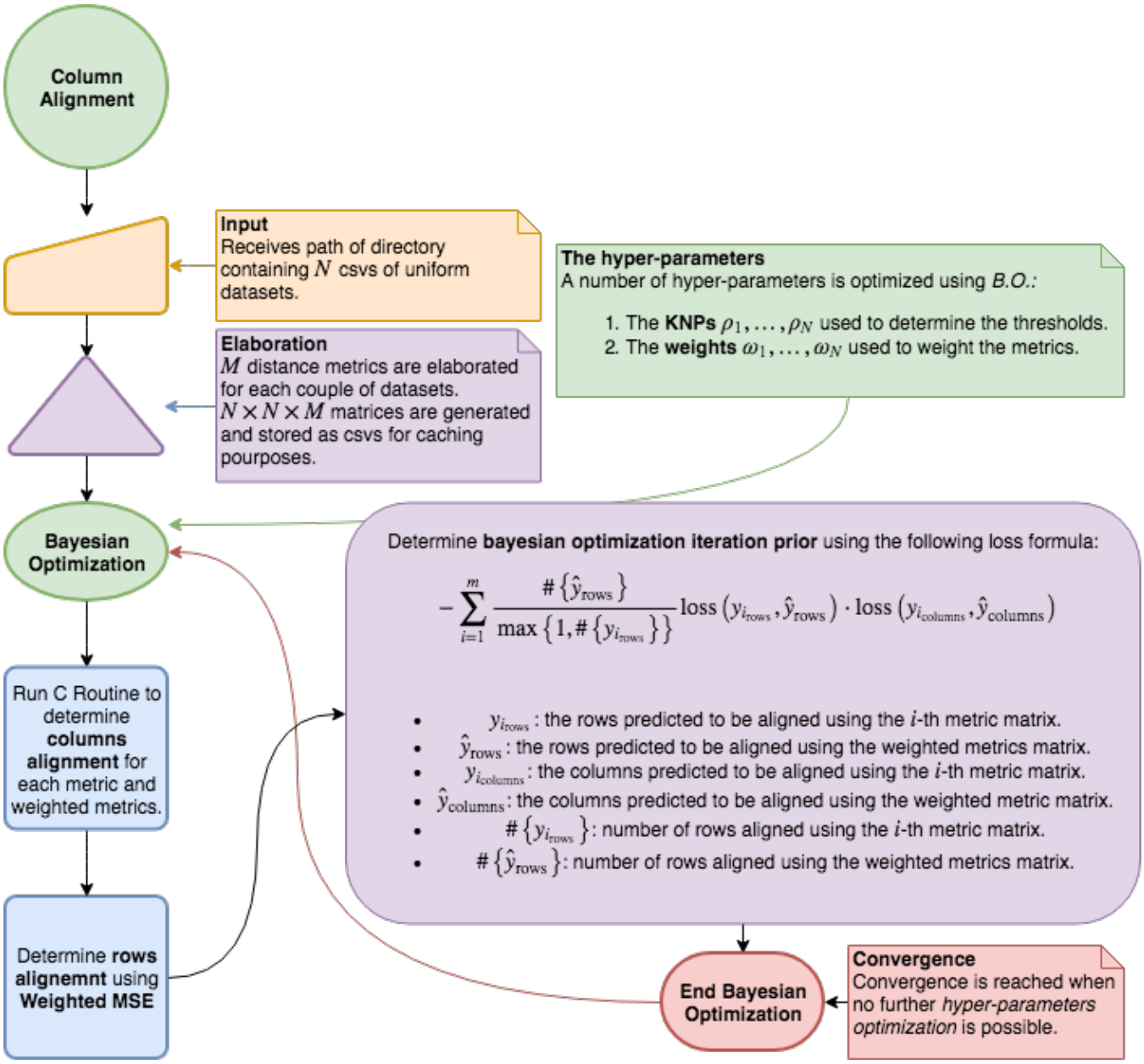
$\#\{\hat{y}_{\textbf{rows}}\}$**:** number of rows aligned by the metrics matrix.

2.3 Process visualization

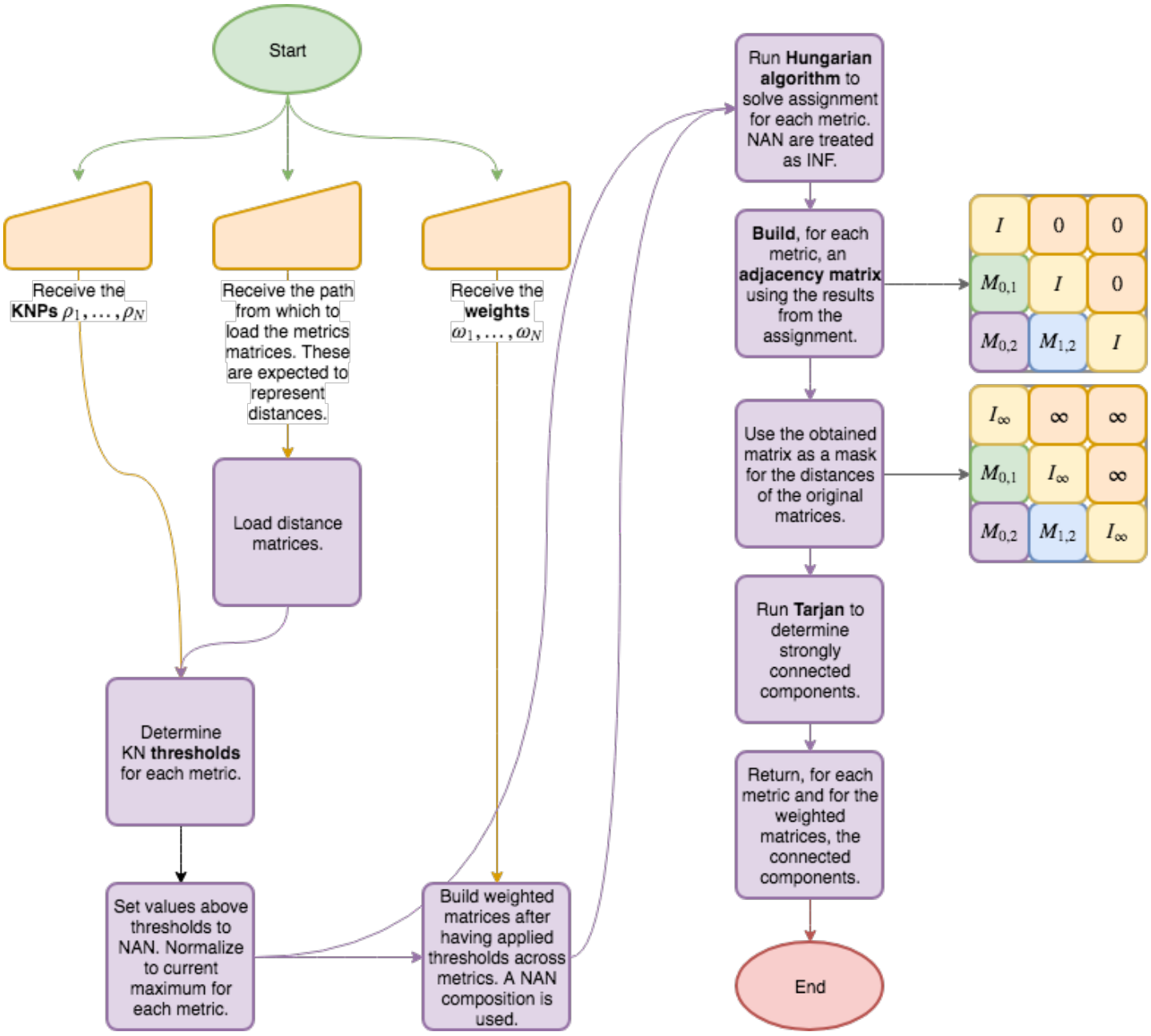
2.3.1 Complete process visualization



2.3.2 Alignment process visualization



2.3.3 C column alignment routine



3

Metrics for columns

What follows is a list of the metrics that are used to align the columns.

3.1 Some useful definitions

Definition 3.1.1 (Pairwise distance). A pairwise distance is any distance implemented so that given two vectors $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ creates a matrix $d\left(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}\right)$ of real values of size $\left(\left|\underline{\boldsymbol{x}}\right|, \left|\underline{\boldsymbol{y}}\right|\right)$ where every value $d_{ij} = d(x_i, y_j)$.

Definition 3.1.2 (TFIDF). In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Definition 3.1.3 (Column name). We call **column name** the header of the column, in the following example C_1, C_2, and in a real example could be *proteins* or *fats*:

	C_1	C_2
R_1	4	0.3
R_3	9	0.6

Tabella 3.1: Dataset example

Definition 3.1.4 ($k$grams). Given a string of characters, its $k$grams are a sliding window of length k.

Definition 3.1.5 (Kolmogorov–Smirnov test). In statistics, the Kolmogorov–Smirnov test (K–S test or KS test) is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test).

Definition 3.1.6 (Mann–Whitney U test). In statistics, the Mann–Whitney U test (also called the Mann–Whitney–Wilcoxon (MWW), Wilcoxon rank-sum test, or Wilcoxon–Mann–Whitney test) is a nonparametric test of the null hypothesis that it is equally likely that a randomly selected value from one sample will be less than or greater than a randomly selected value from a second sample.

Unlike the t-test it does not require the assumption of normal distributions. It is nearly as efficient as the t-test on normal distributions.

3.2 Strings

3.2.1 Pairwise cosine distance on TFIDF of columns names

Once the TFIDF of the columns names is determined, using as dictionary all the columns in every dataset that is to be merged, the cosine distance beetween every TFIDF vector is determied.

3.2.2 Pairwise cosine distance on Average TFIDF of columns content

Once the TFIDF of every textual value in every column is determined, using as dictionary all textual values in every column, we determine the mean of the TFIDF vectors of the cells in each column.

We proceed them to determine the cosine distance beetween every TFIDF vector.

3.2.3 Pairwise Jaccard distance of kgrams on columns names

Once the *k*grams set of each column name is determined, we proceed to determine the pairwise Jaccard distance.

3.2.4 Pairwise Fuzzy Jaccard distance of kgrams on columns content

Once the *k*grams set of each cell is determined, a mean of the set is created and a pairwise fuzzy Jaccard distance is determined.

3.2.5 Mask of Units in columns names

If there are units in the column names we can ensure that columns with different units are not aligned.

3.3 Numericals

3.3.1 Pairwise Kolmorov Smirnov Test

We determine the p-value of the KS test, in a pairwise fashion.

3.3.2 Pairwise Mann Whitney Test

We determine the p-value of the MW test, in a pairwise fashion.

3.3.3 Pairwise Magnitude clustering

For each column, magnitude clusters are determined using the clusters **maximal variance** to determine the number k of clusters for each column.

Columns to be paired need to have at least one magnitude cluster in common.

4

Results and conclusions

4.1 Test data

To test the procedure, the data will be generated from a big table, adding gaussian noise to the values proportionally to the column mean and variance (we cannot apply the same noise to columns whose mean measures in the μg as one in the kg for example, as real world data with the same data do not experience this kind of variance).

Column names and textual column contents will be noised up using a dictionary of synonims made for this purpose.

4.2 Results and Conclusions

It was expected for the bayesian process to identify a set of weights such that the various metrics had to concur on the column alignment: by the **axiom of Condorcet**, a number of decision makers should achieve a better answer than any of them alone.

The metrics are, tough, either too noisy (the non-parametric tests for example have little value with less than 2000 datapoints) or have an extremely strong signal (tfidf distances either align or not names): since a subset of "decision makers" yielded extremely similar answers (tough not completely correct) and the other had noisy answers, the bayesian optimizer simply killed off the noisy metrics by assigning them weights and KNP extremely close to zero and to the others extremely close to one instead.

The **landscape** determined by the proposed score function, therefore, resulted into an extremely flat one, with little to no optimization further possible after having killed off the noisy metrics.

The conclusion seems straightforward: the metrics used were too noisy to determine a formula for alignment.

4.2.1 A positive note

The only semi-supervised aspect of the project was the selection of *known negative percentage* for each metric based on the metric run on the same dataset, which was an addition to previous projects, has been successful to determine a soft threshold for the values that simply aren't related to those that might be, more than an approach using a mean of a percentage of the maximum value.