Designing generalisation evaluation function through human-machine dialogue
Patrick Taillandier, Julien Gaffuri

To cite this version:
Patrick Taillandier, Julien Gaffuri. Designing generalisation evaluation function through human-machine dialogue. Sixth international conference on Geographic Information Science - GIScience 2010, Sep 2010, Zurich, Switzerland. 6 p. hal-02752919

HAL Id: hal-02752919
https://hal.inrae.fr/hal-02752919
Submitted on 3 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Designing generalisation evaluation function through human-machine dialogue

Patrick Taillardier\textsuperscript{1}, Julien Gaffuri\textsuperscript{2}

\textsuperscript{1}IRD, UMI UMMISCO 209, 32 avenue Henri Varagnat, 93143 Bondy, France
Email: patrick.tailliardier@gmail.com

\textsuperscript{2}IFT, MSI, UMI 209, ngo 42 Ta Quang Buu, Ha Noi, Viet Nam

\textsuperscript{3}IGN – COGIT laboratory – Paris-Est University
73 avenue de Paris, 94165 Saint-Mandé cedex, France
Email: julien.gaffuri@ign.fr

1. Introduction
A classic approach in automated generalisation consists in formalising generalisation as an optimisation problem: the goal is to find a state of the data that maximises an evaluation function that is supposed to assess the generalisation state of the data, according to the user need (e.g. Wilson et al., 2003). A key issue of this approach concerns the design of this evaluation function. Unfortunately, designing such a function remains a difficult task. Indeed, while the final user of the generalised data can easily describe his/her need in natural language, it is often far more difficult to express his/her expectations in a formal language that can be used by generalisation systems.

In this paper, we propose an approach dedicated to generalisation evaluation functions design. An evaluation function previously designed by a user is improved through a dialogue between the user and a generalisation system. The idea is to collect user preferences by letting him/her compare different generalisation results for a same object.

In Section 2, the context of this work is introduced. Section 3 is devoted to the presentation of our approach. Section 4 describes an experiment carried out for building generalisation and Section 5 concludes.

2. Context

2.1 Automated evaluation of generalisation results
If many works focus on the generalisation process automation, only a few deal with automatic evaluation of generalisation outcomes. A classic approach consists in evaluating the generalisation quality by means of a set of constraints translating the expectation towards the generalisation (Beard, 1991). The constraint assessment is often represented by a numeric satisfaction value. The overall generalisation is evaluated by aggregating all the constraint satisfaction values. If the computation of individual constraint satisfaction values is often well-managed, the definition of the aggregating function remains complex (Bard, 2004). This paper focuses on this problem.
3.2. Initialization of the Composition Set

Our approach is composed of 3 steps that are described in the next sections. Our approach is composed of a set of binary relations that will be used to define the composition set. Each binary relation is associated with a weight that reflects the importance of the relation. The weights are calculated using a formula that takes into account the frequency of the relations in the dataset.

The first step is the construction of the composition set, which is done using the following formula:

\[ \text{composition set} = \{ \text{relations} \mid \text{weight} \geq \text{threshold} \} \]

3.1. General approach

3. Proposed approach

We propose an approach to design the composition function based on the following principles:

- The role of a relation in the composition set is determined by its weight. The higher the weight, the more important the relation is.
- The evaluation function is defined as follows:

\[ \text{evaluation function} = \sum_{i=1}^{n} w_i \cdot (\text{relation}_i) \]

Where \( w_i \) is the weight associated with relation \( i \), and \( (\text{relation}_i) \) is the score assigned to relation \( i \).

2.2. Design of an evaluation function

This approach allows us to design an evaluation function that is sensitive to the frequency of relations in the dataset. The frequency of a relation is represented by a numeric value, which is calculated by counting the number of times the relation appears in the dataset. This value is then used to calculate the weight of the relation using a formula that takes into account the frequency of the relation in the dataset.

2. Formulation of the evaluation function design

This approach is designed to provide a flexible and adaptable solution to the problem of composing relations. It allows us to design an evaluation function that is sensitive to the frequency of relations in the dataset, and to adjust the weights of the relations based on their importance.

A classical approach to solve this problem consists in using supervised machine learning techniques. These techniques consist in training a model from examples. The model is then used to predict the composition of a new relation based on the properties of the relations in the dataset. This approach is useful for problems where the relations are not predefined and need to be learned from the data.

The evaluation function design is a complex problem which was studied in various research works. Several approaches for predicting the composition of relations were proposed, but few proposed general approaches for predicting the composition of relations are described in this paper.
3.3 Capture of the user preferences

The second step concerns capture of the user preferences: comparisons are successively presented to the user, who gives his/her preference for each of them. This sequence is reiterated until a specific number of comparisons have been presented to the user.

For each comparison between two generalisations $A$ and $B$, the user can choose:

- Generalisation $A/B$ is far better than Generalisation $B/A$
- Generalisation $A/B$ is better than Generalisation $B/A$
- Generalisation $A/B$ is slightly better than Generalisation $B/A$
- Generalisation $A$ and Generalisation $B$ are equivalent

Figure 1 presents the comparison interface of the developed prototype.

![Module for User Need Definition](image)

Fig. 1. The comparison interface

3.4. Evaluation function definition

The last step consists in learning an evaluation function from the captured user preferences: the parameter values (i.e. the constraint weights $w_i$ and the power $p$) that best fits the preferences given by the user during the previous step are computed. We propose to formulate this problem as a minimisation problem. We define a global error function that represents the inadequacy between an evaluation function (and thus the parameter values assignment) and the user preferences. Our goal is to find the parameter values that minimise the global error function.

Let $f_{eval}(gen)$ be the current evaluation function that evaluates the quality of a generalisation $gen$; $c_{gen1,gen2}$ a comparison between two generalisations, $gen_1$ and $gen_2$; $p_c$ the user preference for the comparison $c$. We define the function $comp(c, f_{eval}, p_c)$ that determines for a comparison $c$ if the user preference $p_c$ is compatible with the evaluation function $f_{eval}$, i.e. if the preference is consistent with the quality order obtained by applying the evaluation function. $comp(c, f_{eval}, p_c)$ is computed as follows:
The initial evaluation function was designed by an expert of the AGENT model. We defined two sets of 100 different propositional rules (the learning and the testing set). The design goal is to develop a model that generates a consistent with the acquisition of the experts. The AGENT model has been the core of numerous descriptive works and is used as a neural network in several machine learning algorithms. However, the objective of the constraint with a set of constraints (Bjork, 1999). The problem of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search. The constraint with a set of constraints (Bjork, 1999). The problem of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search (Gloverer, 1999). The procedure of this kind of algorithms is to start with an initial search.
4.2. Results and discussion

Table 1 presents the results on the two comparison sets. It shows for each evaluation function and comparison sets the global error (c.f. Section 3.4).

|          | Initial function | Learn function |
|----------|------------------|----------------|
| Learning set | 44.1%            | 27.4%          |
| Test set   | 40.1%            | 29.0%          |

Table 1. Results

These results reveal that the learnt function has allowed an improvement of the global error: for both learning and testing sets, the global errors of the initial function are higher than for the learnt function. However, the quality improvement after the use of the method is only of 11% for the test set. An explanation is the lack of constraints (for example, an orientation constraint). For example, when a comparison composed of two building generalisations, which differ only in term of orientation is shown, the user always prefers the one whose orientation is close to the building initial orientation. Because there is no orientation constraint taken into account into the evaluation function, the difference of the two generalisations can not be measured by the system, and the reason of the different assessment by the user remains ignored. In this context, our approach, through an examination of the incompatible comparisons, can help to determine some important missing constraints and identify faulty ones.

5. Conclusion

In this paper, we presented an approach dedicated to the definition of a generalisation evaluation function. We proposed a method based on a human-machine dialogue and the capture of user preferences on generalisation samples. An experiment carried out in the domain of cartographic generalisation showed how our approach can help users to define better evaluation functions.

This work is at its beginning. In the near future, we plan to carry out more experiments, in particular to study the impact of the initial evaluation function on the results.

Our long-term purpose is to provide a method to learn the user preferences concerning all objects and group of objects contained in its data. The last stage of this research would be to automatically learn a user final evaluation method for a complete map piece. Such a system would be able to make an automatic interview of the user, allowing him to give his specific requirements for all characteristics of the map.

References

Bard S, 2004, Quality Assessment of Cartographic Generalisation, Transactions in GIS, 8, 63-81.
Barrault M, Regnauld N, Duchêne C, Haire K, Baeijs C, Demazeau Y, Hardy P, Mackaness W, Ruas A and Weibel R, 2001, Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalization. ICC, 2110-2116
Beard K, 1991, Constraints on rule formation. In Buttenfield, B. & Mcmaster, R. (eds.) Map generalisation: making rules for knowledge representation, 121-135.
Clancy C, Hecker J, Stuntebeck E and O'Shea T, 2007, Applications of Machine Learning to Cognitive Radio Networks, Wireless Communications, vol. 14, no. 4, 47-52
andMachine Intelligence, 30(8).

IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008. Learning local objective functions using a genetic algorithm. Workshop on Progress in Automatic Map Generalization, Parts.

Wilson II, J., and Ware, J., 2003. Reducing graphic conflict in scale reduced maps. Workshop on Progress in Automatic Map Generalization, Parts.

Hubert P. and Russ A., 2003. A method based on samples to capture user needs for generalization. Journal on Computing.