Spatial classification with Fuzzy Lattice Reasoning

Constantinos Mavridis
Electrical and Computer Eng. Dept
Democritus University of Thrace
Xanthi, Greece

Ioannis N. Athanasiadis
Information Technology Group
Wageningen University
Wageningen, The Netherlands
ioannis@athanasiadis.info

ABSTRACT

This work extends the Fuzzy Lattice Reasoning (FLR) Classifier to manage spatial attributes, and spatial relationships. Specifically, we concentrate on spatial entities, as countries, cities, or states. Lattice Theory requires the elements of a Lattice to be partially ordered. To match such requirement, spatial entities are represented as a graph, whose number of nodes is equal to the amount of unique values of the spatial attribute elements. Then, the graph nodes are linearly arranged to formulate a partially ordered set; and thus be included in the Fuzzy Lattice classifier. The overall problem of incorporating spatial attributes in FLR was deduced to a Minimum Linear Arrangement problem. A corresponding open-source implementation in R has been made available on CRAN repository. The proposed method was evaluated using an open spatial dataset from the National Ambient Air Quality Standards (NAAQS). We investigated whether the addition of the spatial attribute contributed to any improvements in classification accuracy; and how linear arrangement alternatives may affect it. Experimental results showed that classification accuracy is above 85% in all cases, and the use of spatial attributes resulted to an increased accuracy of 92%. Alternative linear arrangements did not contribute significantly in improving classification accuracy in this case study.

CCS CONCEPTS

• **Computing methodologies → Vagueness and fuzzy logic**; **Machine learning**; *Spatial and physical reasoning*; • **Information systems →** *Geographic information systems*;

KEYWORDS

Fuzzy Lattice Reasoning, classification, spatial data, spatial data mining, lattice theory, spatial classification, linear arrangement

ACM Reference Format:
Constantinos Mavridis and Ioannis N. Athanasiadis. 2018. Spatial classification with Fuzzy Lattice Reasoning. In *Proceedings of International Conference on Internet of Things and Machine Learning, Liverpool, United Kingdom, October 17–18, 2017 (IML'17)*, 7 pages.
https://doi.org/10.1145/3109761.3158378

IML'17, October 17–18, 2017, Liverpool, United Kingdom

ACM ISBN 978-1-4503-5243-7/17/10…$15.00
https://doi.org/10.1145/3109761.3158378

1 INTRODUCTION

In the Internet of Things Era, Big Data are collected in unprecedented velocity and volume. A very common characteristic of Internet of Things data is that they often include spatial attributes, let them be locations of smart devices, sensors, mobile phones or users. Big Data and Internet of Things pose new challenges for machine learning algorithms, including managing and extracting knowledge from spatial and temporal data. Spatial data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from spatial databases [16].

Conventional data mining techniques are not directly suitable for extracting spatial patterns, due to the complexity of spatial data and intrinsic spatial relationships. Efficient tools are needed for extracting information from spatial data are crucial to organizations which make decisions based on large spatial datasets, spread across many application domains including ecology and environmental management, public safety, transportation, earth sciences, epidemiology and climatology [18]. In contrast to traditional approaches on numeric or nominal data, spatial data mining algorithms are required to discover neighbor relationships to extract useful patterns. This is necessary based on the assumption that neighboring objects significantly influence each other [5].

In this work, we concentrate on the Fuzzy Lattice Reasoning classifier[1, 2, 11], and propose a method for including spatial data attributes in the formation of Fuzzy Lattice Rules, by linearly arranging spatial attributes to form (fuzzy) lattices.

The rest of the paper is structured as follows. In Section 2 we present shortly the basic definitions of Fuzzy Lattice theory and the principles of the Fuzzy Lattice Reasoning Classifier: the Fuzzy Lattice Rule engine, rule induction and decision making. Then spatial data attributes in the FLR framework are introduced by representing spatial data in the form of graphs, and then linearly arranging them to form fuzzy lattices. A reference implementation of the method is provided as open-source software available on CRAN. In Section 3 an experimental evaluation using a real dataset from GeoCommons is presented, and the FLR Classifier parameters are tuned to find the best performance. Results show significant performance by including the spatial attributes. The paper concludes with a discussion about the findings and future work.

2 METHODS

While in the literature there is some work on modelling spatial relations and operations using partially ordered sets [13], this is the first time that spatial data attributes are used as constituent lattices in a product lattice, and are investigated for reasoning in the context of the Fuzzy Lattice Reasoning classifier[11].

This section first summarizes useful mathematical definitions about fuzzy lattices [1, 2, 4, 6, 10–12] and the Fuzzy Lattice Reasoning classifier [1, 2, 11], and then introduces a novel representation for spatial data attributes in the framework of fuzzy lattices.

2.1 Fuzzy Lattices

A partially ordered set is called **lattice** L, when any two of its elements a, b have a greatest lower bound $(a \wedge b)$ and a least upper bound $(a \vee b)$. A lattice L is **complete** when each of its subsets has both a least upper bound and a greatest lower bound also in L. The least element its typically denoted as O and the greatest element as I. Two lattice elements $a, b \in \mathsf{L}$ are *comparable*, when $a \leq b$, or $b \leq a$ or are incomparable $(a||b)$.

The Cartesian product of N constituent lattices is called a **product lattice**, and is also a lattice [4]. The greatest lower bound of two product lattice elements $x = (x_1, \ldots, x_N)$ and $y = (y_1, \ldots, y_N)$ is $x \wedge y = (x_1 \wedge y_1, \ldots, x_N \wedge y_N)$, and the least upper bound is $x \vee y = (x_1 \vee y_1, \ldots, x_N \vee y_N)$ [4, 6]. A product lattice could combine disparate types of data as constituent lattices, including as vectors of real numbers, propositions, events in a probability space, intervals, sets, or graphs [11].

A lattice L together with a membership function $\mu : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ such that $a \leq b \iff \mu(a, b) = 1$ is called a **fuzzy lattice** [12]. The framework of fuzzy lattices has been applied in various settings and applications for decision-making [10, 12]. In this work we approach the fuzzy lattice framework from a rule-based perspective, by extending the Fuzzy Lattice Reasoning classifier, presented in [11]. Key elements of the fuzzy lattice framework are presented shortly here, and the reader may refer to previous works for a more in depth presentation [1, 2, 4, 6, 10–12].

Any real function $v : \mathsf{L} \rightarrow \mathsf{R}$ defined on a lattice L is called a **valuation function** if it satisfies the condition $v(m) + v(n) = v(m \wedge n) + v(m \vee n)$ for all elements m, n of L. A **positive** valuation function additionally satisfies $a < b \iff v(a) < v(b)$ [4].

An **inclusion measure** $\sigma : \mathsf{L} \times \mathsf{L} \rightarrow [0, 1]$ is defined on a complete lattice L so that that for each $m, n, x \in \mathsf{L}$ the following conditions are satisfied [11]:

$$\sigma(m, O) = 0, \forall m \neq O \tag{1}$$

$$\sigma(m, m) = 1 \tag{2}$$

$$m < n \Rightarrow \sigma(x, m) < \sigma(x, n) \tag{3}$$

$$m \wedge n < m \Rightarrow \sigma(m, n) < 1 \tag{4}$$

Any lattice L for which an inclusion measure σ can be defined is a fuzzy lattice with σ the membership function. If there is a positive valuation function v defined on a lattice L where $v(O) = 0$, this is a sufficient condition for defining two inclusion measures

$$k(a, b) = \frac{v(b)}{v(a \vee b)} \tag{5}$$

$$s(a, b) = \frac{v(a \wedge b)}{v(a)} \tag{6}$$

and both $\langle \mathsf{L}, k \rangle$ and $\langle \mathsf{L}, s \rangle$ are fuzzy lattices [10].

The only requirement for specifying a fuzzy lattice is the selection of an appropriate positive valuation function v on a lattice, for which $v(O) = 0$. Any kind of partially ordered data that can define a lattice, as numbers, sets, graphs becomes available in the framework of fuzzy lattices if a positive valuation function is ascribed to it [1].

Also note that the framework of fuzzy lattices has been extended to lattices of closed intervals [12]. A closed interval of lattice L elements is defined as $[m, n]$, for every $x \in \mathsf{L}$ that satisfies $m \leq x \leq n$. The set of all closed intervals of lattice elements in L is also a complete lattice with upper bound $[O, I]$ and lower bound $[I, O]$, and an ordering relation is defined as: $[a, b] \leq [c, d] \equiv \{c \leq a, b \leq d\}$.

Given two elements $[a, b]$ and $[c, d]$ of a lattice of closed intervals, the least upper bound and the and the greatest lower bound can be respectively defined as:

$$[a, b] \vee [c, d] = [a \wedge c, b \vee d] \tag{7}$$

$$[a, b] \wedge [c, d] = \begin{cases} [a \vee c, b \wedge d], \text{if} \quad a \vee c \leq b \wedge d) \\ 0, \text{otherwise} \end{cases} \tag{8}$$

How to define a valuation function v_τ for the lattice of closed intervals has been discussed in [12, 15]. Based on the positive valuation function v defined on a lattice L the valuation function in the lattice of closed intervals is $v_\tau([a, b]) = v(\theta(a)) + v(b)$ [12].

Also, both the inclusion measures k and s defined on a fuzzy lattice in equations 5, 6 can be applied on the lattice of closed intervals as [11, 12, 15]:

$$k_\tau([k, l], [m, n]) = \frac{v_\tau([k, l])}{v_\tau([k, l] \vee [m, n])} \tag{9}$$

$$s_\tau([k, l], [m, n]) = \frac{v_\tau([k, l] \wedge [m, n])}{v_\tau([k, l])} \tag{10}$$

Based on the above, both k_τ, and s_τ can be used as membership functions for defining fuzzy lattices of closed intervals, and have been used for defining a fuzzy lattice rules in the context of the Fuzzy Lattice Reasoning (FLR) Classifier. Note that the inclusion measure k_τ is more usable compared to s_τ [1].

2.2 Fuzzy Lattice Reasoning (FLR) Classifier

Many data structures of practical interest are lattice ordered. The Fuzzy Lattice Reasoning Classifier [1, 2, 11] is a classifier for inducing a rule-based inference engine from data, based on the instruments of the fuzzy lattice framework presented above. Below we summarize the how the fuzzy lattice rule engine operates.

A **fuzzy lattice rule** $a \rightarrow c$ consists of an element a of a fuzzy lattice $\langle \mathsf{L}, \mu \rangle$ (premise or antecedent), and a label $c \in \mathsf{C}$ (conclusion or consequent). The inclusion measure μ of the fuzzy lattice defines the **degree of truth** for deriving to the rule consequent of against the perception x. Using the Eq. 5 above: $\mu(x, a) = k(x, a) = \frac{v(a)}{v(x \vee a)}$. This definition applies to the whole framework of fuzzy lattices, and includes lattices of closed intervals and product lattices, also combining disparate data types as constituent lattices. This is of particular interest as it allows for combining spatial data with non-spatial data attributes in a single rule.

A **fuzzy lattice rule engine** $\mathcal{E}_{\langle \mathsf{L}, \mu \rangle, \mathsf{C}}$ is as a set of commonly activated fuzzy lattice rules $\mathcal{E}_{\langle \mathsf{L}, \mu \rangle, \mathsf{C}} \equiv \{a_i \rightarrow c_i\}$, where $a_i \in \langle \mathsf{L}, \mu \rangle$, $c_i \in \mathsf{C}$, and $i = 1 \ldots M$. The rules of a fuzzy lattice rule engine compete to each other to perform reasoning, and the rule with the highest degree of truth prevails.

For example, consider a fuzzy lattice rule engine that contains three rules:

$$\mathcal{E} = \begin{cases} a_1 \rightarrow c_1 \\ a_2 \rightarrow c_2 \\ a_3 \rightarrow c_2 \end{cases} \tag{11}$$

where $a_{1...3}$, are elements of a fuzzy lattice $\langle \mathsf{L}, \mu \rangle$ and $\{c_{1...2}\}$ two predefined classes (labels). Note that the last two rules point to the same class c_2.

When an input element $x \in \mathsf{L}$ is presented to the engine $\mathcal{E}$, then the engine will compute the degree of truth table for all three rules, as: $c_1 = \mu(x, a_1)$, $c_2 = \mu(x, a_2)$, and $c_3 = \mu(x, a_3)$. The fuzzy lattice reasoning engine will respond with the class $c_i = maxarg(\mu(x, a_i))$.

The task of inducing a fuzzy lattice rule engine is commonly referred to as **training**. Let a training set of $u_{1...K}$ elements of lattice L, each one of which is associated with a class label $c \in \mathsf{C}$. The training pairs are presented to a fuzzy lattice rule engine $\mathcal{E}$, with the objective to induce a set of fuzzy lattice rules associating any object $u_{1...M}$ with the corresponding classification label.

All instances of the training set can be treated as fuzzy lattice rules. Each element of the training set can be consider to formulate a fuzzy lattice rule $u_i \rightarrow c_i$, where u_i is the antecedent and c_i the consequent. This enables to induce a naive fuzzy lattice rule classifier directly from the training set, simply by memorizing all training instances as fuzzy lattice rules. The corresponding naive fuzzy lattice rule engine will consist out of M (trivial) rules

$$\mathcal{E}_{trivial} = \begin{cases} u_1 \rightarrow c_1 \\ \dots \\ u_i \rightarrow c_i \\ \dots \\ u_M \rightarrow c_M \end{cases} \tag{12}$$

Previous work [11] proposed a training process that joins lattice rules pointing to the same class and thus formulates lattice rules of higher size. This is performed in a single pass iteration over all training instances, as follows:

FLR training algorithm (simplified from [11])

Step-0: Let a fuzzy lattice rule engine $\mathcal{E}$ that consists of R rules. All rules are considered to be *"set"*. The engine could be initially empty.

Step-1: Present the next training element in the form of a fuzzy lattice rule $u_i \rightarrow c_i$ to the initially *"set"* rules of the engine.

Step-2: If no more rules in $\mathcal{E}$ are *"set"*, then append a new rule $u \rightarrow c$ to the engine $\mathcal{E}$ and go to Step-1.

Step-3: Compute the truth table of all the *"set"* rules in $\mathcal{E}$ against the antecedent u. The rule $a_J \rightarrow c_J$, that produces the highest value in the truth table, is considered as a candidate winner.

Step-4: If $c = c_J$ and the size of $u \vee a_J$ is less than a predefined threshold (vigilance parameter, ρ_{crit}), then replace the candidate winner rule in the engine with: $u \vee a_J \rightarrow c_J$. Go to Step-1.
Else, remove the candidate winner rule from the *"set"* ones and go to Step-2.

Note that the vigilance parameter ρ_{crit} lies in the interval $[0.5, 1]$ independently of the number of dimensions in a product lattice [12]. Thus in our experiments below, we used ρ_{crit} as it is not related to the lattice dimension.

In the **testing phase**, a perception x is presented to a fuzzy lattice rule engine $\mathcal{E}$ with the intention to assign it to a class $c_t \in \mathsf{C}$.

Decision making involves the competition of all the rules of the engine in an iterative process. The perception x is presented to all engine rules and the truth table is calculated for each one of them. The element x is assigned to the category the rule $a_J \rightarrow c_J$ with the highest degree of truth, i.e. x is included the most in a_J.

In previous work, the capacity of the FRL engine was further broaden by introducing non-linear positive valuation functions [1, 11], which we will not further describe in this paper.

A fuzzy lattice reasoning classifier can be induced in any domain of partially ordered objects, as that of R^N, graphs or sets. While there has been some research on modelling spatial relations and operations with partially ordered sets [13], to our knowledge this is the first time spatial data attributes are used in the framework of fuzzy lattices for training fuzzy lattice rule engines.

2.3 Spatial data attributes in the framework of Fuzzy Lattices

The main challenge in this work was to employ a representation for spatial data attributes that allows them to be included on the FRL classifier. Specifically, we are interested in spatial attributes in the form of geolocation names, such as countries, regions, cities, etc. Such attributes may be considered forming a graph, whose nodes are the spatial attributes elements. Without loss of generality we consider in this work working with polygons, but this work extends also to lines and points.

The overall problem can be deduced to the Minimum Linear Arrangement problem (MinLA). The Minimum Linear Arrangement (MinLA) problem is a combinatorial optimization problem formulated originally by [8]. The goal is to find a linear ordering of the nodes of a given graph, such that the sum of the weighted edge lengths is minimized [17] (Fig. 1). Essentially, linear arrangement can be seen as permutations of the nodes, and the MinLA is the one that minimizes the sum of the weighted edge lengths [14].

The function we created as a solution to this problem calculates all possible paths (permutations) within the graph. After that, taking into consideration the distance between the nodes, the overall distance of each path is determined and the shortest path is chosen. A preprocessing function then replaces the location names with the number corresponding to the position of each location within the chosen path, giving us a linear arrangement which then can be used directly by the FLR Classifier.

A common choice for spatial frameworks is to model spatial relationships among locations using a contiguity matrix. A contiguity matrix represents a neighborhood relationship defined using adjacency, Euclidean distance, or other metrics [3]. In our case, we used the Euclidean distance between the centroids of the polygons representing regions. Also, we set in the contiguity matrix distances between non-neighboring locations with a very high value, using this penalty to force the algorithm to ignore them unless no other connections are available.

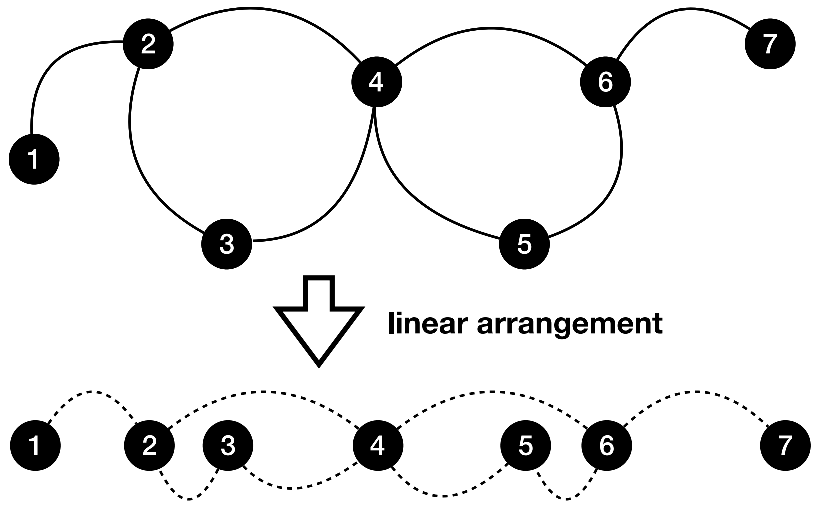


Figure 1: The nodes of a graph (top) and one possible linear arrangement of them (bottom).

Let us describe the process with a practical example. Consider nine US States shown highlighted in Fig. 2a. A graph representation of those states is shown in Fig. 2b, representing each state at the centered set of latitude and longitude coordinates. Center coordinates of each state in this examples have been calculated as the average latitude and longitude of all the zip codes within each state, as reported in [9]. The corresponding continuity matrix can be derived by calculating the Euclidean distance between adjacent state centroids, and illustrated in Table 1. Non neighboring states were assigned an arbitrary high value.

2.4 Implementation and software

The purpose of this work is to extend the FLR Classifier by incorporating spatial attributes. The FLR Classifier was extended and now includes a new preprocessing step that arranges linearly the objects of the spatial attributes.

The linear arrangement of the spatial attributes elements along with other tasks such as removing elements of missing class, separating the data in training and testing etc. occur during the data preprocessing phase. During the training phase, fuzzy lattice rules are created in the form of a FLR engine, which in turn will compete over instances of unknown class in the testing phase. Finally, the user needs to review the results of the process such as the lattice rules extracted or the overall accuracy of the predictions made.

A reference implementation of FLR is available in the WEKA Knowledge Analysis Environment written in Java[1]. While our initial intention was to add the new feature to an updated version of FLR that will still be part of the WEKA environment, after further consideration it was decided that it would be more efficient to choose a different platform, as it would be extremely difficult to incorporate a new datatype in WEKA.

[1]And stored in the WEKA SVN Repository

Thus, the R programming environment was chosen, due to its popularity among generalist data scientists, and also because its effectiveness in data handling, extensibility and the breadth of available packages. The development of this package was completed in three stages. At first, the FLR Classifier was re-implemented in R language while keeping the functionality of the original algorithm intact. In the second stage, the geo-preprocessing functionalities were developed, i.e. spatial attributes are linearly arranged,. Finally, certain requirements set by CRAN had to be met, in order to share our code. CRAN is a network around the world that stores identical up-to-date versions of code and documentation for R submitted packages. The final software code is available online under GPL license: https://cran.r-project.org/web/packages/FLR/index.html.

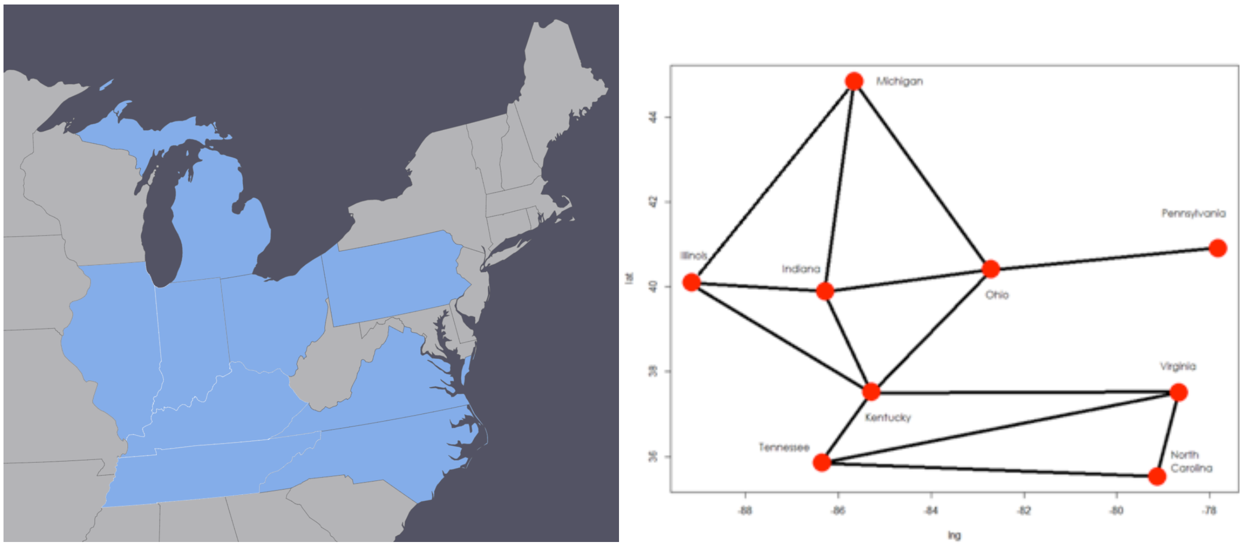
3 EXPERIMENTS AND RESULTS

3.1 Data

In our experiments we used a public spatial dataset reporting the Ozone concentrations violations for several monitoring locations in the US according to the National Ambient Air Quality Standards (NAAQS) for the years 2009-2014. The original dataset was posted on GeoCommons.com [7], under a Creative Commons Attribution 3.0 License. The dataset attributes include:

- the number of days above the NAAQS, for each year
- the fourth highest daily max value, for each year
- the design values, as computed using Federal Reference Method for each three-year period
- the completeness of the data collected by monitoring station

and the design value status label that classifies monitoring sites as either attaining (A) or violating (V). We filtered the dataset to include only nine states, namely Illinois, Indiana, Kentucky, Michigan, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia. The filtered dataset consists of 296 locations reported. The filtered dataset is included in our open source implementation deposited on CRAN.



(a) (b)

Figure 2: An example of nine US states. (a) Map of the east USA highlighting the nine states. From amcharts.com. (b) Graph representation of the nine states. Each state is depicted at the centroid of latitude and longitude coordinates

Table 1: The contiguity table for the 9 US states used in the example. Values are distances between centroids of adjacent states.

	IL	IN	KY	MI	NC	OH	PA	TN	VA
IL		3.522	5.679	7.210					
IN	3.522		3.145	6.096		4.419			
KY	5.679	3.145				4.738		2.425	8.114
MI	7.210	6.096				6.513			
NC								8.849	2.493
OH		4.419	4.738	6.513			6.012		
PA						6.012			
TN			2.425		8.849				9.626
VA			8.114		2.493			9.626	

3.2 Experimental configuration

Finding the best FLR model for the classification task described above, involves tuning two parameters: the positive valuation function type (linear or sigmoid), and the vigilance parameter ρ_{crit}. For evaluating the FLR spatial extension, we introduce one more parameter to tune: how the linear arrangement of the spatial attributes is performed. In our experiments, we performed 10-cross fold validation and and repeated five times, while ρ_{crit} was given the values 0.5, 0.6, 0.7. The default sigmoid function parameters where used and not further tuned.

The purpose of the experiment is to evaluate how linear arrangement of the spatial attributes may affect decision making and how much the inclusion of a spatial attribute may improve the classification. Thus, we treated spatial attribute preprocessing in three different ways: (a) Using no spatial attribute arrangement; (b) use the overall minimum linear arrangement; (c) use a local minimum linear arrangement, by defining the starting node.

3.3 Results and discussion

Table 2 summarizes the average classification accuracy across all folds and repetitions, for the different parameters for which FLR classifier was tuned. Accuracy for each parameter combination is at least 85.9% with the higher accuracy being 92%. Comparing the

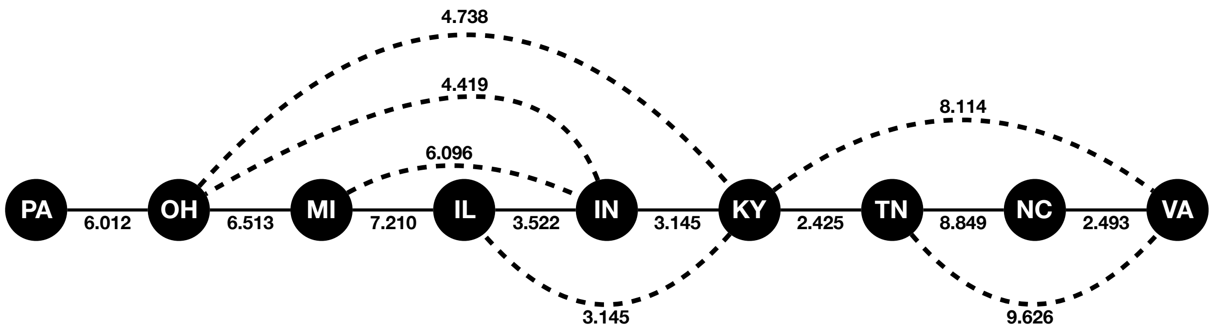


Figure 3: The selected linear arrangement of the nine states; minimizing the total distance of the selected path. Next to the arcs are illustrated the costs from the contiguity matrix. The total length (cost) for the illustrated path is 40.169.

Table 2: Classification accuracy results. Average values reported across all folds and repetitions. Standard deviation reported in brackets

Spatial data attribute linear arrangement	Linear			Sigmoid		
	ρ_{crit} = 0.5	ρ_{crit} = 0.6	ρ_{crit} =0.7	ρ_{crit} = 0.5	ρ_{crit} = 0.6	ρ_{crit} = 0.7
None	**85.93 (6.22)**	86.00 (6.13)	88.83 (4.54)	**86.21 (6.19)**	86.55 (6.08)	88.69 (6.86)
Overall	85.93 (6.22)	86.21 (6.97)	90.41 (4.94)	86.21 (6.19)	88.55 (5.97)	**92.00 (4.75)**
1	85.93 (6.22)	86.21 (6.97)	90.41 (4.94)	86.21 (6.19)	88.69 (5.83)	**92.00 (4.75)**
2	85.93 (6.22)	**90.14 (5.70)**	90.41 (4.53)	86.21 (6.19)	**90.14 (6.19)**	90.00 (5.36)
3	85.93 (6.22)	88.97 (5.95)	90.41 (6.00)	86.21 (6.19)	89.45 (5.84)	90.97 (4.17)
4	85.93 (6.22)	88.28 (6.82)	90.21 (4.48)	86.21 (6.19)	88.69 (5.83)	90.97 (4.87)
5	85.93 (6.22)	86.55 (7.01)	90.28 (5.34)	86.21 (6.19)	87.31 (6.25)	91.10 (6.04)
6	85.93 (6.22)	87.31 (5.76)	90.55 (5.16)	86.21 (6.19)	88.55 (5.72)	89.59 (6.01)
7	85.93 (6.22)	88.21 (6.12)	90.48 (5.19)	86.21 (6.19)	88.62 (5.71)	90.48 (4.85)
8	85.93 (6.22)	87.31 (5.76)	**90.62 (5.07)**	86.21 (6.19)	87.66 (6.19)	90.35 (5.17)
9	85.93 (6.22)	87.31 (5.76)	90.62 (5.07)	86.21 (6.19)	88.55 (5.72)	89.59 (6.01)

performance of using the linear versus the sigmoid valuation function, the latter provides better results in most cases, although the average improvement is 0.34%.

For the smallest ρ_{crit} value (0.5) the inclusion or not of the spatial attribute does not contribute to any performance difference. The use of the sigmoid valuation function increases the overall accuracy by almost 1%. By increasing the rho_{crit} parameter we observe that the classification accuracy increases. Note that the time required for training increases too.

The inclusion of the spatial attribute in all cases improved the performance independently of the linear arrangement selected. The overall MinLA path offered the best performance for $rho_{crit} = 0.7$, which was 92%, an improvement of more than 3%. Though we observed that the overall MinLA does not offer the best performance improvement in the case of $\rho_{crit} = 0.6$.

4 CONCLUSIONS AND FUTURE WORK

In this paper we investigated linear arrangements of graph elements for including spatial attributes in the framework of the fuzzy lattices, and incorporated it in the Fuzzy Lattice Reasoning classifier [11]. The method proposed offers a tool that exploits implicit information within spatial attributes of a dataset. We tested our method with a real dataset, and in our experiments the inclusion of the spatial attribute improved the decision-making capacity of FLR. The solution we provide was implemented in R programming language, and is offered as an open source repository on CRAN. A number of experiments were executed, in which FLR parameters (valuation function, vigilance parameter) were tuned, and in all cases the inclusion of the spatial attribute resulted in classification accuracy improvements.

The new feature we added is a preprocessing function, which is called before the training phase of the FLR classification process and linearly arrange the elements of the spatial attribute(s).

While the introduced linear arrangement of the spatial feature always improves performance, an interesting finding that worths future investigation is that the linear arrangement of the overall minimum path does not always offer the best performance improvement.

Supplementary material

The software developed for this work as an R package is available at the CRAN repository under GPL license. https://cran.r-project.org/web/packages/FLR/index.html

To install type: `install.packages(ŕFLRŕ)`

Author contributions

INA: Conception and supervision of the work; CM: Data collection, software implementation and simulations; CM, INA: Data analysis and interpretation; CM, INA: Writing the article

REFERENCES

[1] I. N. Athanasiadis. 2007. The Fuzzy Lattice Reasoning Classifier for mining environmental data. In *Computational Intelligence Based on Lattice Theory*, V. G. Kaburlasos and Gerhard X. Ritter (Eds.). Springer-Verlag, Chapter 1, 175–193. https://doi.org/10.1007/978-3-540-72687-6

[2] I. N. Athanasiadis and V. G. Kaburlasos. 2006. Air quality assessment using Fuzzy Lattice Reasoning (FLR). In *IEEE Intl Conf on Fuzzy Systems, World Congress on Computational Intelligence (WCCI 2006)*. IEEE, Vancouver, British Columbia, Canada, 231–236. https://doi.org/10.1109/FUZZY.2006.1681690

[3] T. C. Bailey and A. C. Gatrell. 1995. *Interactive spatial data analysis*. Vol. 413. Longman Scientific & Technical, Harlow, UK.

[4] G. Birkhoff. 1967. *Lattice Theory*. Colloquium Publications, Providence, RI, USA.

[5] S. Chawla, S. Shekhar, Weili Wu, and U. Ozesmi. 2001. Modeling spatial dependencies for mining geospatial data. In *Proceedings of the 2001 SIAM International Conference on Data Mining*. SIAM, SIAM, Chicago, IL, USA, 1–17. https://doi.org/10.1137/1.9781611972719.27

[6] B. A. Davey and H. A. Priestley. 1990. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, UK.

[7] GeoCommons. 2011. Dataset No 356891. (2011). http://geocommons.com/datasets?id=356891

[8] L. H. Harper. 1964. Optimal assignments of numbers to vertices. *J. Soc. Indust. Appl. Math.* 12, 1 (1964), 131–135.

[9] InkPlant. 2011. List of Latitudes and Longitudes for Every State. (2011). https://inkplant.com/code/state-latitudes-longitudes

[10] V. G. Kaburlasos. 2006. *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory: Computational Intelligence and Soft Computing Applications*. Studies in Computational Intelligence, Vol. 27. Springer, Heidelberg, Germany.

[11] V. G. Kaburlasos, I. N. Athanasiadis, and P. A. Mitkas. 2007. Fuzzy Lattice Reasoning (FLR) classifier and its application for ambient ozone estimation. *International Journal of Approximate Reasoning* 45, 1 (May 2007), 152–188. https://doi.org/10.1016/j.ijar.2006.08.001

[12] V. G. Kaburlasos and V. Petridis. 2000. Fuzzy lattice neurocomputing (FLN) models. *Neural Networks* 13, 10 (2000), 1145–1170.

[13] W. Kainz, M. J. Egenhofer, and I. Greasley. 1993. Modelling spatial relations and operations with partially ordered sets. *International Journal of Geographical Information Systems* 7, 3 (1993), 215–229.

[14] J. Petit. 2003. Experiments on the minimum linear arrangement problem. *Journal of Experimental Algorithmics (JEA)* 8 (2003), 2–3.

[15] V. Petridis and V. G. Kaburlasos. 1999. Learning in the Framework of Fuzzy Lattices. *IEEE Transactions on Fuzzy Systems* 7, 4 (1999), 422–440.

[16] J.F. Roddick and M. Spiliopoulou. 1999. A bibliography of temporal, spatial and spatio-temporal data mining research. *ACM SIGKDD Explorations Newsletter* 1, 1 (1999), 34–38.

[17] H. Seitz. 2010. *Contributions to the minimum linear arrangement problem*. Ph.D. Dissertation. Ruprecht-Karls-Universität Heidelberg.

[18] S. Shekhar and S. Chawla. 2003. *Spatial databases: a tour*. Prentice Hall, Upper Saddle River, NJ.