
A nearest neighbour classifier based on probabilistically/possibilistically intervals' number for spam filtering

Yazdan Jamshidi

Young Researchers and Elite Club,
Kermanshah Branch, Islamic Azad University,
Kermanshah, Iran
Email: y.jamshidi@gmail.com,

Abstract: Today, e-mail has become one of the fastest and most economical forms of communication in modern life. However, the increase in e-mail users has resulted in a significant boosting in unsolicited e-mails, widely known as spam, during the past few years. This paper presents an application of Interval's Number KNN (INKNN) for spam filtering. The INKNN algorithm was described lately as a lattice data domain extension of KNN classifier. In our experiment a spam e-mail was presented in the metric space of lattice ordered Interval's Number. Indeed a population of spam e-mails was presented by an Interval's Number. Then INKNN classifier was employed distinguish spam e-mails from non-spam. To investigate the effectiveness of our methods, we conduct extensive experiments on SpamAssassin public mail corpus. Experimental results show that the proposed model is able to achieve higher performance in comparison with those from a number of state-of-the-art machine learning techniques published in the literature.

Keywords: lattice theory; nearest neighbour; spam filtering; intervals' number.

Reference to this paper should be made as follows: Jamshidi, Y. (2016) 'A nearest neighbour classifier based on probabilistically/possibilistically intervals' number for spam filtering', *Int. J. Soft Computing and Networking*, Vol. 1, No. 1, pp.4-16.

Biographical notes: Yazdan Jamshidi received his BS in Applied Mathematics from Shahid Rajaee University (Tehran-Iran) in 2005. He graduated from Shahid Bahonar University (Kerman-Iran) in 2011 with an MS in Computer Science – minor in AI. His interests include artificial neural networks, evolutionary computation, pattern recognition, machine learning, fuzzy logic and granular computing.

1 Introduction

Lattice computing (LC) is an interesting topic which has been taken into account by several authors. The term LC was introduced recently by Graña (2008). More specifically, LC was defined as the class of algorithms that use lattice theory either to achieve pattern recognition or to produce generalisations. Graña and his colleagues have applied LC to image analysis (Graña et al., 2010, 2009a). Moreover, they proposed an endmember threshold selection algorithm (ETSA) (Graña et al., 2009b). Lattices are

popular in mathematical morphology including image processing applications (Graña et al., 2010; Ritter and Wilson, 1996). Neural networks whose computation is based on lattice algebra are known as morphological neural networks. Moreover, algebraic lattices have been used for modelling associative memories (Ritter and Urcid, 2007). In Kohonen (1972) the problem of capacity storage limitation in associative memories (Hopfield, 1984; Kosko, 1989) has been eliminated by proposing bidirectional lattice associative memories. The notion of a fuzzy lattice was proposed by Nanda in 1989 on the basis of the concept of a fuzzy partial order relation (Nanda, 1989). Fuzzy lattices have also been used in clustering and classification algorithms. More specifically, independently from the development of morphological neural networks, Kaburlasos and Petridis (1997; Sussner and Esmi, 2009) have found inspiration in lattice theory and versions of the ART model and have devised another successful approach to lattice-based computational intelligence. Hence, they proposed a fundamentally new and inherently hierarchical approach in neuron-computing named fuzzy lattice neuro-computing (FLN) (Kaburlasos and Petridis, 1997). Moreover, fuzzy lattice reasoning (FLR) classifier was announced for inducing descriptive, decision-making knowledge (rules) in a mathematical data domain including space R^N and it has been successfully applied to a variety of problems such as ambient ozone estimation (Kaburlasos et al., 2007) as well as air quality assessment (Athanasiadis and Kaburlasos, 2006) Sussner and Esmi (2009) have introduced the morphological perceptron with a fusion of fuzzy lattices for competitive learning. A practical advantage of lattice theory is the ability to model both uncertain information and disparate types of lattice-ordered data. Since information granules are partially/lattice-ordered, therefore, LC is proposed for dealing with them (Kaburlasos and Petridis, 1997; Kaburlasos, 2010; Sussner and Esmi, 2011; Kaburlasos et al., 2012).

Especially successful, within LC, was the employment of lattice of fuzzy intervals' number (FINs) to represent populations of samples/measurements (Kaburlasos, 2004). Various interpretations can be proposed for a FIN. For instance, a FIN may be considered as a conventional fuzzy number or it may be interpreted as a possibility distribution or as a probability distribution. Generally speaking, a FIN may be regarded as an information granule for dealing with ambiguity (Kaburlasos, 2006). Note that recently, Papadakis and Kaburlasos (2010) have employed the term intervals' number (IN) instead of the term FIN.

During the last century, the communication capabilities of humanity have been widely improved. In particular, one of the most important steps forward in the human communication domain was the genesis and popularisation of internet, emerged from a packet switching military network (Roberts, 1978). Today, e-mail is a fundamental tool for business communication and modern life, spam represents a serious threat to user productivity and IT infrastructure worldwide. While it is difficult to quantify the level of spam currently sent, many reports suggest it represents substantially more than half of all e-mails sent and predict further growth for the foreseeable future (Espiner, 2005; Radicati Group, 2004; Zeller, 2005). More recent estimates of spam traffic can be found on Kaspersky lab spam statistics report. Several solutions have been proposed to overcome the spam problem. Among the proposed methods, much interest has focused on the machine learning techniques including support vector machines (Cristianini and Shawe-Taylor, 2000) naïve Bayes (Androutsopoulos et al., 2000) neural networks (Wang, 2006), k-nearest neighbour (Crawford et al., 2004), memory-based approach (Sakkis et al., 2003), active learning (Jinlong et al., 2010), boosting of C4.5 (Schapire and Singer, 2000) and C4.5 with PART (Hidalgo et al., 2000). Most of these approaches try to

minimise the number of errors in the classifier and their common concept is that they do not require specifying any rules explicitly to filter out spam mails. Instead, a set of training samples is needed. Perhaps the most straightforward classifier in machine learning techniques is the nearest neighbour classifier. It assigns a test sample the label associated with a majority vote of its nearest neighbours.

This paper presents a nearest neighbour classification algorithm for spam filtering based on probabilistically/possibilistically interpreted INs and lattice theory. A practical advantage of lattice theory is the ability to model both uncertain information and disparate types of lattice-ordered data (Kaburlasos and Petridis, 1997). Indeed, our proposed algorithm is capable of dealing with disparate type of data including real vectors, fuzzy sets, symbols, graphs, images, waves and even any combination of the aforementioned data. It can handle both points and intervals. Learning in the proposed algorithm is carried out fast therefore, in many application, when the data is so massive is the analysis process so time consuming, the proposed algorithm can be a proper choice. The main advantage of employing INs is the accommodation of granular data. The experimental results demonstrate the effectiveness of our proposed model.

The layout of this paper is as follows. In Section 2 the mathematical background on lattices is reviewed. Section 3 explains our proposed model. Section 4 provides empirical results that demonstrate the performance of *INKNN*. Finally, Section 5 summarises the results of this work.

2 Mathematical background

A lattice (L, \preceq) is a partially ordered set (or simply, poset) such that any two of its elements $a, b \in L$ have a greatest lower bound $a \wedge b = \inf \{a, b\}$ and a least upper $a \vee b = \sup \{a, b\}$ bound. The lattice operations \wedge and \vee are also called meet and join, respectively. A lattice (L, \preceq) is called complete when each of its subsets has a least upper bound and a greatest lower bound in L (Birkhoff, 1967). A non-void complete lattice has a least element and a greatest element denoted by O and I , respectively. The inverse \succ of an order relation \preceq is itself an order relation. The order \succ is called the *dual* order of \preceq symbolically \preceq° or \succ . Note that, in this work, we use, ‘straight’ symbols \vee, \wedge and \leq , for real numbers whereas ‘curly’ symbols \wedge, \vee and \preceq are employed for other lattice elements.

A lattice (L, \preceq) can be Cartesian product of N constituent lattices L_1, \dots, L_N i.e., $L = L_1 \times \dots \times L_N$. The lattice operations meet and join of product lattice are defined as below:

$$a \vee b = (a_1, \dots, a_N) \vee (b_1, \dots, b_N) = (a_1 \vee b_1, \dots, a_N \vee b_N) \quad (1)$$

$$a \wedge b = (a_1, \dots, a_N) \wedge (b_1, \dots, b_N) = (a_1 \wedge b_1, \dots, a_N \wedge b_N) \quad (2)$$

A valuation on a crisp lattice L is a real-valued function $v: L \rightarrow R$ which satisfies $v(a) + v(b) = v(a \vee b) + v(a \wedge b)$, $a, b \in L$. A valuation is called monotone iff $a \preceq b$ in L implies $v(a) \preceq v(b)$ and positive iff $a < b$ implies $v(a) < v(b)$.

Consider the set R of real numbers. It turns out that $(\bar{R} = R \cup \{-\infty, +\infty\}, \leq)$ under the inequality relation \leq between $a, b \in R$ is a complete lattice with the least element $-\infty$ and the greatest element $+\infty$ (Kaburlasos and Pachidis, 2011). A lattice (L, \preceq) is totally ordered if and only if for any $a, b \in L$ either $a \succeq b$ or $a < b$. The lattices (\bar{R}^N, \preceq) and $([0, 1]^N, \preceq)$ under inequality relation are not a totally ordered lattice.

Definition 1. Generalised interval is an element of the product lattice $(\bar{R}, \leq^\delta) \times (\bar{R}, \leq) \equiv (\bar{R} \times \bar{R}, \geq \times \leq)$ which is denoted, simply, by (Δ, \preceq) (Papadakis and Kaburlasos, 2010).

A generalised interval will be denoted by $[a, b]$. We remark that (Δ, \preceq) is a lattice with the ordering relation, lattice join and meet defined as below:

$$[a, b] \preceq [c, d] \equiv c \preceq a \quad \text{and} \quad b \preceq d \quad (3)$$

$$[a, b] \vee [c, d] = [a \wedge c, b \vee d] \quad (4)$$

$$[a, b] \wedge [c, d] = [a \vee c, b \wedge d] \quad (5)$$

The set of positive (negative) of generalised intervals $[a, b]$ characterised by $[a \preceq b$ ($a \succ b$) is denoted by $\Delta_+(\Delta_-)$. We remark that (Δ_+, \preceq) is a poset, called poset of positive generalised intervals. For lattice (L, \preceq) we define the set of (closed) intervals as $\tau(L) = \{[a, b] \mid a, b \in L \text{ and } a \preceq b\}$. Augmenting a least (the empty) interval, denoted by $[I, O]$, to $(\tau(L), \preceq)$ leads to a complete lattice $(\tau_O(L), \preceq) = (\tau(L) \cup \{[I, O]\}, \preceq)$. In case of $L = \bar{R}$, the lattice $(\tau_O(L), \preceq)$ is equal to conventional intervals (sets) in \bar{R} . The poset (Δ_+, \preceq) is isomorphic to the poset $(\tau_O(\bar{R}), \preceq)$ i.e., $(\Delta_+ \cup \{O\}, \preceq) \cong (\tau_O(\bar{R}), \preceq)$. Note that an isomorphic function φ from poset P to poset Q is a map, if both ' $x \preceq y$ in $P \Leftrightarrow \varphi(x) \preceq \varphi(y)$ in Q ' and ' φ is onto Q '. Our particular interest here is in the complete lattice $(\tau_O([0, 1]), \preceq)$ with greatest element $[0, 1]$ and least element $[1, 0]$. Due to the aforementioned isomorphism, we employ isomorphic lattices $(\Delta_+ \cup \{O\}, \preceq)$ and $(\tau_O(\bar{R}), \preceq)$, interchangeably.

Proposition 1: A positive valuation function $v: L \rightarrow R$ in a lattice (L, \preceq) implies a metric $d: L \times L \rightarrow R^{\geq 0}$ given by $d(x, y) = v(x \vee y) - v(x \wedge y)$, $x, y \in L$ (Birkhoff, 1967; Rutherford, 1965).

It should be mentioned that the goal of positive valuation function v is to deal with lattice elements. Choosing a suitable valuation function is problem dependent. Various positive valuation functions (Kaburlasos et al., 2007; Liu et al., 2011; Khezeli and Nezamabadi-pour, 2012) have been proposed in the literature.

Based on the positive valuation function v of lattice (L, \preceq) and an isomorphic function $\theta: (L, \preceq^{\circ}) \rightarrow (L, \preceq)$ a valuation function v_{Δ} in $(\Delta_+ \cup \{O\}, \preceq)$ is defined as:

$$v_{\Delta}([a, b]) = v(\theta(a)) + v(b) \quad (6)$$

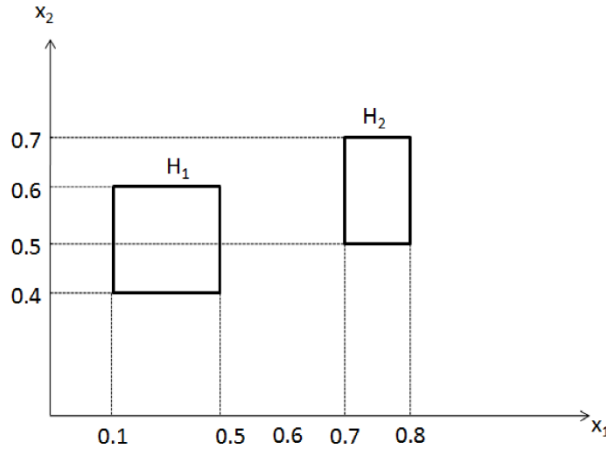
As a consequence the distance between two intervals in lattice $(\Delta_+ \cup \{O\}, \preceq)$ is computed as follows (Kaburlasos, 2006):

$$d_{\Delta}([a, b], [c, d]) = v_{\Delta}([a, b] \vee [c, d]) - v_{\Delta}([a, b] \wedge [c, d]) \quad (7)$$

For two N-dimensional hypercubes $A = [a_1, b_1] \times \dots \times [a_N, b_N]$ and $B = [c_1, d_1] \times \dots \times [c_N, d_N]$ the following metric distance between two intervals A and B is defined:

$$d_{\Delta}([a, b], [c, d]) = \sum_{i=1}^N [v(\theta(a_i \wedge c_i)) - v(\theta(a_i \vee c_i)) + v(b_i \vee d_i) - v(b_i \wedge d_i)] \quad (8)$$

Figure 1 Two two-dimensional hypercubes H_1, H_2



Example: The corresponding distance between two hypercubes H_1, H_2 shown in Figure 1 using the positive valuation function $v(x) = x$ is calculated as follows:

$$\begin{aligned} d_{\Delta}(H_1, H_2) &= d_{\Delta}([0.1, 0.5], [0.7, 0.8]) + d_{\Delta}([0.4, 0.6], [0.5, 0.7]) \\ &= [(0.9 - 0.3) + (0.8 - 0.5)] + [(0.6 - 0.5) + (0.7 - 0.6)] = 1.1 \end{aligned}$$

Definition 2: Generalised intervals' number (GIN) is a function $G: (0, 1] \rightarrow \Delta$ (Kaburlasos, 2006).

Let G denote the set of GINs. Since (G, \preceq) represents the Cartesian product of complete lattices (Δ, \preceq) . Hence, (G, \preceq) is a complete lattice.

Definition 3: An intervals' number, or IN for short, is a GIN F such that both $F(h) \in (\Delta_+ \cup \{O\}, \preceq)$ and $h_1 \leq h_2 \Rightarrow F(h_1) \succcurlyeq F(h_2)$ (Kaburlasos, 2006).

An IN F can be written as the set union of generalised intervals; in particular, $F = \cup_{h \in (0, 1]} \{[a_h, b_h]\}$, where both interval-ends a_h and b_h are functions of $h \in (0, 1]$. A point entry $x_i \in R$ is represented by the trivial IN $x_i = \cup_{h \in (0, 1]} \{[x_i, x_i]\}$. Various interpretations can be proposed for an IN. For instance, an IN may be considered as a conventional fuzzy number or it may be interpreted as a possibility distribution or as a probability distribution. Generally speaking, an IN may be regarded as an information granule for dealing with ambiguity. It has been shown that (F, \preceq) is a lattice (Kaburlasos, 2004, 2006).

Proposition 2: Let F_1 and F_2 be INs in the lattice (F, \preceq) of INs. Assuming that the following integral exists, a metric function $d_F: F \times F \rightarrow R^{\geq 0}$ is given by

$$d_F(F_1, F_2) = \int_0^1 d_\Delta(F_1(h), F_2(h)) dh.$$

Note that a Minkowski $d_p: F^N \times F^N \rightarrow R^{\geq 0}$ can be defined between two N-tuple Ins $F_1 = [F_{1,1}, \dots, F_{1,N}]^T$ and $F_2 = [F_{2,1}, \dots, F_{2,N}]^T$ as $d_p(F_1, F_2) = [d_F^p(F_{1,1}, F_{2,1}) + \dots + d_F^p(F_{1,N}, F_{2,N})]^{1/p}$. In our experiments we have set $p = 2$.

In order to compute the above integral first, both INs F_1 and F_2 should be represented by their α -cuts (intervals) for different values of h in the interval $[0, 1]$. Second, for each aforementioned value of h , the metric between the corresponding α -cuts is calculated. Finally, the average of the aforementioned obtained metrics is calculated. In this work we have used 32 values of h .

For further information about Interval's number we refer the reader to (Papadakis and Kaburlasos, 2010).

3 The proposed model

This section presents a nearest neighbour classifier called *INKNN* based on lattice theory for spam filtering. Note that *INKNN* stands for Interval's Number K -nearest neighbour. The idea behind K -nearest neighbour algorithm is quite straightforward. It assigns to a test sample a class label of its closest neighbour using a metric distance such as Euclidean distance typically used in conventional nearest neighbour. *INKNN* classifier operates on the metric product lattice (F, \preceq) , where F denotes the set of conventional

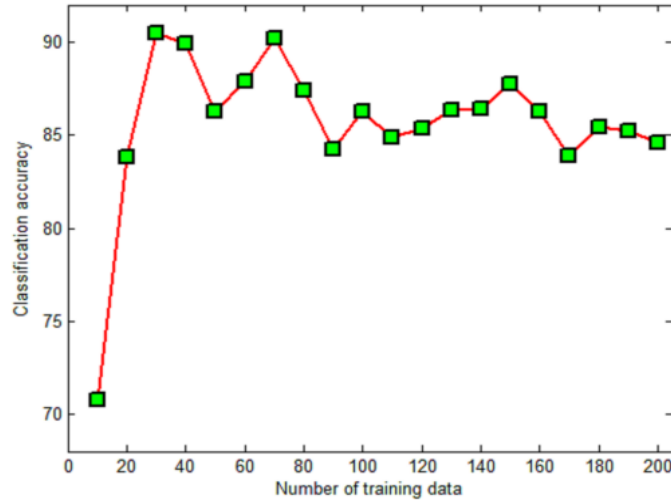
interval-supported convex fuzzy sets. Indeed, it can deal with any pair of fuzzy sets with arbitrary-shaped membership functions and computes a unique distance for any pair of fuzzy sets. That is, it can cope with disparate type of data including real vectors, fuzzy sets, symbols, graphs, images, waves and even any combination of the aforementioned data and this shows the ability of the algorithm in combining different type of data. The algorithm for $K = 1$ is described in the following:

The proposed algorithm

- 1 Store all labeled training data $\{(F_1, C_1), \dots, (F_n, C_n)\}$ where, $F_i \in F^N$, $C_i \in L$, $I \in \{1, \dots, n\}$ and L represents the set of class labels.
- 2 Consider a new unlabeled datum $F_0 \in F^N$ for classification.
- 3 For each class C_i , $i = 1, \dots, n$ Compute the distance $d_p(F_0, F_i)$
- 4 Let $J = \operatorname{argmin}\{d_p(F_0, F_i)\}$, $i \in \{1, \dots, n\}$.
- 5 The class label of F_0 is defined to be C_J given that $d_p(F_0, F_J) \leq \mu_0$, where μ_0 is a user defined threshold otherwise; the label 'unknown' is assigned to F_0 .

The above algorithm calculates the distance of a new unlabeled datum F_0 from all the labelled training data and then F_0 is assigned to the nearest sample provided that the corresponding distance is less than a user defined threshold μ_0 otherwise F_0 is treated as an 'unknown' pattern.

Figure 2 Relation between size of INs and the classification accuracy (see online version for colours)



It should be mentioned that an IN may include any number of data. Note that an IN may even include a single datum x . In the latter case, the corresponding IN as it was mentioned before, is represented by trivial intervals $[x, x]$ for all h in $[0, 1]$. The number of INs required for a class depends on the data. For instance, if all the data in a class are both near to each other and far apart from data in different classes, then one IN per class is enough. However, if the data in a class are in clusters, with data from different classes' in-between, then more than one IN per class is needed. The exact number of INs is not known 'a priori', Only the data can decide the numbers of required INs. This matter also

holds for the number of data included in the INs. Figure 2 plots the relation between size of INs and the classification accuracy for the SpamAssassin public mail corpus. Note that the size of an IN is defined as the number of training data samples used to induce an IN by algorithm CALCIN (Papadakis and Kaburlasos, 2010).

It reveals that accommodating more data into an IN does not necessarily increase the classification accuracy because different samples of an IN may belong to different clusters and this causes deterioration in the percent of correct classification. In particular Figure 1 shows that, the INKNN achieved its highest classification accuracy for accommodating number of 30 training data samples in each IN. By increasing the number of data samples the accuracy is decreased to approximately 87% is again the classification accuracy improves to nearly 90% and finally, providing more data to the INs causes the classification accuracy to fluctuate in the range of 83–87%. The fluctuation of the accuracy is typically due to the intermingling of the two classes.

4 Experimental results

4.1 Data pre-processing

There are several known and well-defined collections of ham and spam messages and many researchers use them as a basis in their comparisons (Zorkadis et al., 2005). In this work we have employed the proposed model to the Spam Assassin Public Corpus. The SpamAssassin public mail corpus is a selection of 1,897 spam messages and 4,150 legitimate e-mails. In our experiment we have randomly selected 1,277 spam e-mails and 2,723 legitimate messages.

Many e-mails would contain similar types of entities (e.g., numbers, other URLs, or other e-mail addresses), the specific entities (e.g., the specific URL or specific dollar amount) will be different in almost every e-mail. Therefore, one method often employed in processing e-mails is to normalise these values, so that similar components of an e-mail such as URLs are treated the same. The following pre-processing and normalisation steps have been made to the data set:

- *lower-casing*: the capitalisation is ignored by converting the entire e-mail into lower case
- *stripping HTML*: all HTML tags are removed from the e-mails
- *normalising URLs*: All URLs are replaced with the text ‘httpaddr’
- *normalising e-mail addresses*: all e-mail addresses are replaced with the text ‘emailaddr’
- *normalising numbers*: all numbers are replaced with the text ‘number’
- *normalising dollars*: all dollar signs (\$) are replaced with the text ‘dollar’
- *word stemming*: words are reduced to their stemmed form. For instance, ‘invest’, ‘invested’, ‘investing’ are all replaced with ‘invest’
- *removal of non-words*: non-words and punctuation have been removed. All white spaces such as spaces, tabs and newlines have all been trimmed to a single space character.

The aforementioned steps of processing data typically improve the performance of a spam classifier. For instance Normalising URLs has the effect of letting the spam classifier make a classification decision based on whether any URL was present, rather than whether a specific URL was present. Since each e-mail contains a list of words thus, after pre-processing the e-mails a vocabulary list should be prepared. That means, to choose which words should be used in our classifier. Our vocabulary list was selected by choosing all words which occur at least a 100 times in the spam corpus, resulting in a list of 1,899 words (<http://www.spamassassin.apache.org/publiccorpus/>; <https://www.class.coursera.org/ml/class/index>). Figure 3 shows a sample spam e-mail and the result of the pre-processing steps is shown in Figure 4.

Figure 3 A spam e-mail

Do You Want To Make \$1,000 Or More Per Week? If you are a motivated and qualified individual – I will personally demonstrate to you a system that will make you \$1,000 per week or more! This is NOT mlm. Call our 24 hour pre-recorded number to get the details. 000-456-789. I need people who want to make serious money. Make the call and get the facts. Invest two minutes in yourself now! 000-456-789. Looking forward to your call and I will introduce you to people like yourself who are currently making \$10,000 plus per week! 000-456-789.

Figure 4 Pre-processed spam e-mail

do you want to make dollarnumb or more per week if you ar a motiv and qualifi individu i will person demonstr to you a system that will make you dollarnumb number per week or more thi is not mlm call our number hour pre record number to get the detail number number number i need peopl who want to make seriou monci make the call and get the fact invest number minut in yourself now number number number look forward to your call and i will introduc you to peopl like yourself who ar current make dollarnumb number plu per week number number number

4.2 Comparison with previous work

In this section, we evaluate the classification performance of the proposed model and compare the obtained results with those from a number of state-of-the-art machine learning techniques published in the literature. In order to provide a meaningful comparison all the algorithms have been implemented in the same environment using the C++ object oriented programming language, the same partitioning of data sets for training and testing, the same order of input patterns and a full range of parameters. In order to avoid overfitting, words that occur rarely in the corpus were omitted and all words which occur at least a 100 times in the spam corpus are added to our vocabulary list, resulting in a list containing 1,899 words. Thus, our selected dataset consist of 4,000 1,899-dimensional vectors and ten-fold cross validation is used for performance assessment. Indeed, for a faster simulation we have used 10% of the data for training, which is a small value but yet sufficient for trainingis the remaining 90% as the testing set. To compare the learning capability, Table 1 shows the comparison of the

experimental results of the INKNN with the ones produced by the *fuzzy-ART* (Carpenter et al., 1991), *GRNN* (Specht, 1991), *KNN* (Tomek, 1976) and *SVM* (Drucker et al., 1997).

In all our experiments in order to achieve the best performance we have considered *GRNN* for different values of *variance parameter* between 0 and 0.5 in steps of 0.001. For *fuzzy-ART* we have set the *choice parameter* to 0.01 and the values of *vigilance* and *learning parameters* have been adopted between zero and one in steps of 0.1. Moreover, for the *SVM* classifier here we employed linear kernel and sequential minimal optimisation (SMO) method to find the separating hyperplanes. The value of K for conventional *KNN* was set in the range [1, 20]. Typically $K = 10$ produced the best results. The INKNN classifier was applied for $K = 1$, because $K = 1$ gave better results than other values of K in this application. Furthermore we have employed isomorphic function and the positive valuation function $\theta(x) = 1 - x$ and $\nu(x) = x$, respectively.

Table 1 Comparison of classification accuracy for different methods

<i>Accuracy\algorithm</i>	<i>Fuzzy-ART</i>	<i>GRNN</i>	<i>KNN</i>	<i>SVM</i>	<i>INKNN</i>
Average	80.78 (3)	72.53 (5)	80.07 (4)	93.15 (1)	87.35 (2)
Best	85.92 (3)	74.22 (5)	83.61 (4)	95.00 (2)	95.83 (1)

Table 1 summarises the average and best classification accuracy and ranking of each classifier on the testing data. In other words, each table slot, which belongs to a specific classifier, contains the average of classification accuracy on ten runs and the best of ten. The number in brackets in each table slot indicates the ranking of each classifier based on the percentage of correct classification. As can be seen in Table 1, the *SVM* classifier has obtained the first ranking on the average classification accuracy followed by *INKNN* classifier. The *KNN* and *fuzzy-ART* algorithms performed poorly with rankings 4, 3. The *GRNN* has achieved the worst average ranking. Regarding the best classification accuracy among ten experiments, the proposed *INKNN* classifier outperformed all other classifiers and the *SVM* obtained the second ranking. In fact, the *INKNN* has demonstrated here considerable potential in classification, which it is attributed to the fact that all the data in a class, are both near to each other and apart from the data in the other class as explained in Section 3.

5 Conclusions

This paper presents a nearest neighbour classification algorithm for spam filtering based on probabilistically/possibilistically interpreted INs and lattice theory. A practical advantage of lattice theory is the ability to model both uncertain information and disparate types of lattice-ordered data (Kaburlasos and Petridis, 1997). INKNN was employed on the SpamAssassin public mail corpus which is a popular dataset for spam filtering. The experimental results demonstrate the effectiveness of the proposed model.

For future work we plan to accommodate data belonging to the same cluster in each IN using clustering algorithms as well as evaluating the performance for more than One IN per class.

References

- Androutsopoulos, I., Koutsias, J., Chandrinou, K.V. Paliouras, G. and Spyropoulos, C.D. (2000) 'An evaluation of naive Bayesian anti-spam filtering', in Potamias, G., Moustakis, V. and van Someren, M. (Eds.): *Proceedings of Workshop on Machine Learning in the New Information Age*, Barcelona, pp.9–17.
- Athanasiadis, I.N. and Kaburlasos, V.G. (2006) 'Air quality assessment using fuzzy lattice reasoning (FLR)', *Proceedings of IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, 16–21 July, pp.29–34.
- Birkhoff, G. (1967) *Lattice Theory*, 3rd ed., American Mathematical Society Providence, RI.
- Carpenter, G.A., Grossberg, S. and Rosen, D.B. (1991) 'Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system', *Neural Networks*, Vol. 4, No. 6, pp.759–771.
- Crawford, E., Koprinska, I. and Patrick, J. (2004) 'Phrases and feature selection in e-mail classification', *Proceedings of the Ninth Australasian Document Computing Symposium*, Melbourne, Australia.
- Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge, UK
- Drucker, H., Burges, J.C., Kaufman, L., Smola, A.J. and Vapnik, V.N. (1997) 'Support vector regression machines', *Advances in Neural Information Processing Systems 9, NIPS 1996*, pp.155–161, MIT Press.
- Espiner, T. (2005) *Demand for Anti-Spam Products to Increase*, June, ZDNet, UK.
- Graña, M. (2008) 'A brief review of lattice computing', *Proceedings of the World Congress on Computational Intelligence*, Hong Kong, China, 1–6 June, pp.1777–1781.
- Graña, M., García-Sebastián, M. and Hernández, C. (2009a) 'Lattice independent component analysis for fMRI analysis daft slides', *Lecture Notes in Computer Science*, Vol. 5768, pp.725–734.
- Graña, M., Savio, A.M., García-Sebastián, M. and Fernández, E. (2010) 'A lattice computing approach for on-line fMRI analysis', *Image and Vision Computing*, Vol. 28, No. 7, pp.1155–1161.
- Graña, M., Villaverde, I., Maldonado, J.O. and Hernández, C. (2009b) 'Two lattice computing approaches for the unsupervised segmentation of hyperspectral images', *Neurocomputing*, Vol. 72, No. 10–12, pp.2111–2120.
- Hidalgo, J.G., López, M.M. and Sanz, E.P. (2000) 'Combining text and heuristics for cost-sensitive spam filtering', *Proc. of the 4th Computational Natural*.
- Hopfield, J.J. (1984) 'Neurons with graded response have collective computational properties like those of two state neurons', *Proc. of the National Academy of Sciences*, Vol. 81, No. 10, pp.3088–3092.
- Jinlong, W., Ke, G. and Quan, V.H. (2010) 'Spam cooling: a parallel heterogeneous ensemble spam filtering system based on active learning techniques', *J. Convergence Inf. Technol.*, Vol. 5, No. 4.
- Kaburlasos, G.V. and Pachidis, T. (2011) 'A Lattice-Computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application', *Information Fusion*, March 2014, Vol. 16, pp.68–83.
- Kaburlasos, V.G. (2004) 'FINs: lattice theoretic tools for improving prediction of sugar production from populations of measurements', *IEEE Transactions on Systems, Man and Cybernetics – Part B*, Vol. 34, No. 2, pp.1017–1030.
- Kaburlasos, V.G. (2006) 'Towards a unified modeling and knowledge-representation based on lattice theory', *Studies in Computational Intelligence*, Vol. 27, Springer, Heidelberg, Germany.

- Kaburlasos, V.G. (2010) 'Granular fuzzy inference system (FIS) design by lattice computing', in Corchado, E., Graña, M. and Savio, A.M. (Eds.): *Proceedings, Part II, 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS '10)*, San Sebastián, Spain, 23–25 June 2010, LNAI 6077, Springer, pp.410–417.
- Kaburlasos, V.G. and Petridis, V. (1997) 'Fuzzy lattice neurocomputing (FLN): a novel connectionist scheme for versatile learning and decision making by clustering', *International Journal of Computers and Their Applications*, Vol. 4, No. 3, pp.31–43.
- Kaburlasos, V.G., Athanasiadis, I.N. and Mitkas, P.A. (2007) 'Fuzzy lattice reasoning (FLR) classifier and its application for ambient ozone estimation', *International Journal of Approximate Reasoning*, Vol. 45, No. 1, pp.152–188.
- Kaburlasos, V.G., Papadakis, S.E. and Amanatiadis, A. (2012) 'Binary image 2D shape learning and recognition based on lattice computing (LC) techniques', *Journal of Mathematical Imaging and Vision*, Vol. 42, No. 2–3, pp.118–133.
- Khezeli, Y.J. and Nezamabadi-pour, H. (2012) 'Fuzzy lattice reasoning for pattern classification using a new positive valuation function', *Advances in Fuzzy Systems*, pp.1–8, Article ID 206121.
- Kohonen, T. (1972) 'Correlation matrix memory', *IEEE Trans. Computers*, C-21, pp.353–359.
- Kosko, B. (1989) 'Bidirectional associative memories', *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 18, No. 1, pp.49–60.
- Liu, H., Xiong, S. and Fang, Z. (2011) 'FL-GrCCA: a granular computing classification algorithm based on fuzzy lattices', *Computers and Mathematics with Applications*, Vol. 61, No. 1, pp.138–147.
- Nanda, S. (1989) 'Fuzzy lattices', *Bulletin Calcutta Mathematical Society*, Vol. 81, pp.1–2.
- Papadakis, S.E. and Kaburlasos, V.G. (2010) 'Piecewise-linear approximation of non-linear models based on probabilistically/possibilistically interpreted intervals' numbers (INs)', *Information Sciences*, Vol. 180, No. 24, pp.5060–5076.
- Radicati Group (2004) *Anti-Spam 2004 Executive Summary*, Technical report, Radicati Group.
- Ritter, G.X. and Wilson, J.N. (1996) *Handbook of Computer Vision Algorithms in Image Algebra*, CRC Press, Boca Raton.
- Ritter, G.X. and Urcid, G. (2007) 'Learning in lattice neural networks that employ dendritic computing', *Studies in Computational Intelligence (SCI)*, Vol. 67, pp.25–44.
- Roberts, L. (1978) 'The evolution of packet switching', *Proceedings of the IEEE*, Vol. 66, No. 11, pp.1307–1313 [online] <http://www.packet.cc/files/ev-packet-sw.html> (accessed April 2012).
- Rutherford, D.E. (1965) *Introduction to Lattice Theory*, Oliver and Boyd Ltd., Edinburgh, Great Britain.
- Sakkis, G., Androutopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D. and Stamatopoulos, P. (2003) 'A memory-based approach to anti-spam filtering for mailing lists', *Inf. Retr.*, Vol. 6, No. 1, pp.49–73.
- Schapire, R.E. and Singer, Y. (2000) 'BoosTexter, a boosting-based system for text categorization', *Machine Learning*, Vol. 39, Nos. 2/3, pp.135–168.
- Specht, D.F. (1991) 'A general regression neural networks', *IEEE Transactions on Neural Networks*, Vol. 2, No. 6, pp.568–576.
- Sussner, P. and Esmi, E. (2009) 'Introduction to morphological perceptrons with competitive learning', *Proceedings of the International Joint Conference on Neural Networks*, Atlanta, GA, pp.3024–3031.
- Sussner, P. and Esmi, E.L. (2011) 'Morphological perceptrons with competitive learning: lattice-theoretical framework and constructive learning algorithm', *Information Sciences*, Vol. 181, No. 10, pp.1929–1950.
- Tomek, I. (1976) 'An experiment with the nearest neighbor rule', *IEEE Transactions on Information Theory*, Vol. 6, No. 6, pp.448–452.

- Wang, B., Jones, G.J.F. and Pan, W. (2006) 'Using online linear classifiers to filter spam emails', *Pattern Anal. Appl.*, Vol. 9, No. 4, pp.339–351.
- Zeller Jr, T. (2005) 'Law barring junk e-mail allows a flood instead', *The New York Times*, 1 February.
- Zorkadis, V., Panayotou, M. and Karras, D.A. (2005) 'Improved spam e-mail filtering based on committee machines and information theoretic feature extraction', *Proceedings of the International Joint Conference on Neural Networks*, Montreal, Canada, Vol. 18, pp.799–807.