1-1-2015

E-mail authorship attribution using customized associative classification

Michael R. Schmid
Concordia University

Farkhund Iqbal
Zayed University

Benjamin C.M. Fung
McGill University

Digital Investigation 14 (2015) S116–S126

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS 2015 USA

E-mail authorship attribution using customized associative classification

Michael R. Schmid [a], Farkhund Iqbal [b, *], Benjamin C.M. Fung [c]

[a] Concordia Institute for Information Systems Engineering, Concordia University, QC, Canada
[b] College of Technological Innovation, Zayed University, United Arab Emirates
[c] School of Information Studies, McGill University, QC, Canada

Keywords:
Authorship
Crime investigation
Anonymity
Data mining
Associative classification
Writeprint
Rule mining

A B S T R A C T

E-mail communication is often abused for conducting social engineering attacks including spamming, phishing, identity theft and for distributing malware. This is largely attributed to the problem of anonymity inherent in the standard electronic mail protocol. In the literature, authorship attribution is studied as a text categorization problem where the writing styles of individuals are modeled based on their previously written sample documents. The developed model is employed to identify the most plausible writer of the text. Unfortunately, most existing studies focus solely on improving predictive accuracy and not on the inherent value of the evidence collected. In this study, we propose a customized associative classification technique, a popular data mining method, to address the authorship attribution problem. Our approach models the unique writing style features of a person, measures the associativity of these features and produces an intuitive classifier. The results obtained by conducting experiments on a real dataset reveal that the presented method is very effective.

Introduction

E-mail has emerged as one of the most popular means of online communication. Unfortunately, it is often used for sending unsolicited e-mails, conducting phishing scams, and for spreading malware due to the lack of standard security and privacy mechanisms. In many misuse cases, an offender either masks his/her actual identity or impersonates someone of high authority to trick a user into disclosing valuable personal information such as credit card or social insurance numbers. According to the annual report published by the Internet Crime Complaint Center,[1] 16.6% of the total reported 336,655 cybercrimes were e-mail scams called “FBI scams”, in which the attackers pretended to be an FBI official in order to defraud victims.

Most published methods are used as a postmortem panacea and there exists no concrete proactive mechanism for securing e-mail communication (Iqbal et al., 2010a). It has been shown Iqbal et al. (2010a) that analyzing e-mail content for the purpose of authorship analysis can help prosecute an offender by precisely linking him/her to a malicious e-mail with tangible supporting evidence. Most existing authorship techniques (de Vel et al., 2001a,b; Teng et al., August 2004; Zheng et al., 2003) study different stylometric features (e.g., lexical, structural, syntactical, content-specific and idiosyncratic) separately but very few of them have studied the collective effect of these features.

Building a writeprint by combining lexical, syntactical, structural, semantic, and content-specific attributes produces more promising results than when individual features are compared separately. This reveals the importance of

* Corresponding author.
E-mail addresses: michael.schmid@concordia.ca (M.R. Schmid), farkhund.iqbal@zu.ac.ae (F. Iqbal), ben.fung@mcgill.ca (B.C.M. Fung).

[1] http://www.ic3.gov/media/annualreport/2011_IC3Report.pdf.

http://dx.doi.org/10.1016/j.diin.2015.05.012

interdependence, correlation, and associativity of stylometric features on the accuracy of methods. *Frequent pattern mining* (Agrawal et al., 1993), *sequential pattern mining* (Agrawal and Srikant, 1995), and *association rule mining* are studied for analyzing associativity of features ((Fachkha et al., 2012), (Han et al., 2006)). In this paper, we employ *Associative Classification* (*AC*) (Agrawal et al., 1993), based on association rule discovery techniques, for authorship identification. The developed classification model consists of patterns that represent the respective author's most prominent combinations of writing style features.

There are many different implementations of AC, namely Classification based on Associations (CBA) (Liu et al., August 1998), Classification based on Predictive Association Rules (CPAR) (Han and Yin, 2003), Classification-based on Multiple Association Rules (CMAR) (Li et al., 2001), and Multi-class Classification based on Association Rules (MCAR) (Thabtah et al., 2005). Given the need to accurately quantify the match between the various author's writing styles and the anonymous e-mail, we have concentrated our research on *CMAR*. This variation on AC uses a subset of rules as opposed to a single best rule, to determine which class, or author in our case, is the best match.

Below are some of the pertinent contributions of this paper.

- To our knowledge, this is the first application of *AC* to the authorship attribution problem; the experimental results on real-life data endorse the suitability of the presented approach.
- Association rule mining in AC is different than traditional association rule mining; the former investigates the associativity of features with one another as well as with the target predetermined classes, whereas the later is limited to the analysis of the interdependence between features and do not associate them all to a target class. Therefore, extracted association rules reveal feature combinations that are relevant in distinguishing one author from another in authorship identification.
- Each instance in a classification model shows the features that are related, not only to each other but to the class label as well. As a result, the proposed method builds a concise and representative classifier that can serve as admissible evidence to support the identification of the true author of a disputed e-mail.

The rest of this paper is organized as follows: Section 1 provides a literature review on authorship analysis and classification analysis. Section 2 formally defines the authorship attribution problem and the notion of writeprint by *class association rule* (*CAR*) list. Section 3 describes our new data mining approach for modeling a writeprint from transformed semantic content. Section 4 evaluates the accuracy and efficiency of our suggested method on the Enron e-mail dataset.[2] Section 5 brings the paper to a conclusion.

[2] http://www.cs.cmu.edu/~enron/.

Related work

Authorship attribution is studied as a text categorization and classification problem in the literature (de Vel, August 2000). Generally, a classification model is built using the previously written documents of the suspected authors. The author names are used as class labels in the training and testing processes of model development. Unlike authorship verification, which is studied as one-class (Koppel & Schler) and two-class (Iqbal et al., March 2010b) classification problem, modern authorship attribution, which can be better understood by reading Stamatato's survey Stamatatos (March 2009), can be approached as a multi-class classification problem.

There is no single standard predefined set of features that best differentiates the writing style of individual writers, but some studies Grieve (July 2007) have identified the most representative features in terms of accurately classifying anonymous or disputed texts. Punctuation and n-gram features have proven to be highly representative on their own, but the combination of these features was discovered to be even more characteristic. The relative preference for using certain words over others along with their associations is another highly representative feature. Vocabulary richness, fluency in the language and grammatical and structural preferences of individuals are among these important writing style manifestations. Finally, spelling and grammar mistakes and rare word sequences are also quite characteristic of an authors writing style. One comprehensive study on stylistic features presented by Abbasi and Chen (2008) discusses these with sufficient detail.

Most methods require feature selection as an important step towards maximizing accuracy; our algorithm does not require feature selection because unimportant features will not meet the minimum support threshold. In other words, the algorithm itself performs feature selection, simplifying one of the more complex aspects of authorship attribution.

Authorship analysis has been quite successful in resolving authorship attribution disputes over various types of writings (Mendenhall, 1887). However, e-mail authorship attribution poses special challenges due to its characteristics of size, vocabulary and composition when compared to literary works (de Vel et al., 2001a,b). Literary documents are usually large in size, comprising of at least several paragraphs; they have a definite syntactic and semantic structure. In contrast, e-mails are short and usually do not follow well defined syntax or grammar rules. Thus, it is harder to model the writing patterns of their author. Ledger and Merriam (1994) established that authorship attribution would not be very accurate for texts containing less than 500 words, creating the need for better models Iqbal et al. (2010a) able to handle the characteristics inherent in e-mails. Moreover, e-mails are more informal in style and people are not as conscious about spelling or grammar mistakes particularly in these types of communications. Therefore, techniques that are appropriate for literary and traditional works are not always well suited for e-mail authorship attribution problems.

Iqbal et al. (May 2013) have shown that the e-mail authorship attribution problem can be solved by designing

algorithms that deal with the specific challenges related to e-mail authorship analysis. Our research differs from that work by applying a popular data mining technique called associative classification whereas *AuthorMiner* Iqbal et al. (May 2013) concentrated on frequent item sets. Our proposed method boasts improved classification accuracy and performance, as will be shown in detail in Section 4.

A popular classification method, the *Support Vector Machine* (*SVM*), was applied (de Vel, August 2000; Teng et al., August 2004) over a set of structural and stylistic features for e-mail authorship attribution. de Vel et al. (2001a,b) and Corney et al. (2002) were said to have performed extensive experiments and concluded that classification accuracies are lower when the training data set size decreases, when the number of authors increases, and when the average length of documents decreases. This explains the decline in classification accuracy seen when processing documents with e-mail-like characteristics.

de Vel et al. (2001a,b) further found that the performance of SVM was diminished when the number of function words used increased from 122 to 320, contradicting the tenet that SVM supports high dimensionality and leading to the conclusion that increasing the number of features does not improve accuracy. However, it has been proposed Iqbal et al. (2010a) that identifying key combinations of features that are able to differentiate between writing styles of various suspects and removing useless or noisy features can improve accuracy. Recently, Ding et al. (2015) proposed a systematic approach to visualize writeprints by matching n-gram words and syntactic features between the anonymous message and sample messages.

Generally each of the four feature sets are employed independently, which can result in conflicting attributions (de Vel, August 2000). For example, composition style and word usage may change from one structural pattern to another. Previous authorship attribution techniques also suffered from the challenge of considering too great a number of features, making it difficult to isolate the right feature sets to use for any given e-mail set. de Vel (August 2000) have shown that adding meaningless features may decrease the accuracy of classification when a classifier captures these features as noise. Using common or otherwise weak features for classification also damages the justification of evidence for corroborating the finding, creating a legal problem from a technical one. One of our approaches overcomes this limitation by flexibly extracting the evidence (a set of combinations of frequently occurring features) from the data itself and filtering out noise with user-supplied thresholds that are not content or domain specific.

The problem

Let S be the group of suspected authors of an anonymous e-mail e. Let E_i be a relatively large collection of e-mails written by suspect $S_i \in S$. Let V be a set of distinct words in $\cup E_i$. The *problem of authorship attribution* is to find the most plausible author S_a from the suspects S, whose e-mail collection E_a best matches with the stylometric feature items in the malicious e-mail e. Intuitively, an e-mail collection E_i matches e if E_i and e share similar patterns of writing style features in strongly representative combinations. The primary objective of cyber forensic investigators is to automatically and efficiently model the patterns, or *writeprint*, of each suspect. They can then present such patterns as evidence identifying the author of the malicious e-mail e.

In terms of AC, what exactly is a writeprint? Specifically, we want to extract rules derived from patterns that strongly and uniquely represent the writing style of each suspect S_i, but do not embody the writing style of any other suspect S_j, where $i \neq j$. The rest of this section will discuss the pre-processing of e-mails and formally define the notions of frequent patterns, classification rules, and similarity metrics between an e-mail collection E_i and an anonymous e-mail e.

Extracting stylometric features

For each e-mail, we first remove the headers and appended *forward* or *reply* content. E-mails with less than a few sentences or unrelated text attached are not included, as they would not contain sufficient information about an author's writing style.

For numerical features, we normalize them to a value between 0 and 1, and then discretize each normalized feature into a set of intervals, for example, $[0-0.25]$, $(0.25-0.5]$, $(0.5-0.75]$, $(0.75-1]$, based on *equal-frequency discretization*, where each interval contains approximately the same number of records. Each interval is designated as a feature item. The subsequently normalized frequency of features is then compared against these intervals. Intuitively, the writing style of a collection of e-mails E_i written by suspect S_i is a combination of stylometric feature items that frequently occur in e-mails E_i. These frequently occurring patterns are modeled with the concept of *frequent pattern* (Agrawal et al., 1993) or *frequent stylometric pattern* described in Iqbal et al. (2010a).

Definition 2.1. (*Frequent stylometric pattern*). *Let E be an e-mail collection. Let $sup(F)$ be the e-mails in E that contain the stylometric pattern $F \subseteq V$. A stylometric pattern F is a frequent stylometric pattern in E if $sup(F) \geq min\_sup$, where the minimum support threshold min_sup is a positive real number provided by the user.*

The writing style of a suspect S_i is therefore represented as a set of frequent stylometric patterns, denoted by $FP(E_i) = \{F_1, \ldots, F_k\}$, extracted from the set of e-mails E_i. These patterns are used to derive a high quality class association rule list that consists of the frequent stylometric patterns by means of pruning and ranking. The details will be discussed in Section 3.

Associative classification writeprint

Fingerprint identification in forensic science, known as dactyloscopy, is the process of comparing two instances of friction ridge skin impressions to determine whether these impressions could have come from the same individual. In authorship attribution applied by cyber forensic specialists, we can do something similar by identifying a distinguishable writing style, a *writeprint*, of an individual.

Writeprints, as described in this paper, cannot uniquely tell apart every individual on earth, but a properly identified writeprint is accurate enough to meaningfully distinguish the writing style of an individual from a small group of suspects, given a sufficient quantity of their previously written texts. In this subsection, we will describe *AC* and writeprint modeling.

The task of classification in general is to build a classifier from a training data set to accurately classify each test record from a test data set. There are many different approaches used for classification, such as decision tree (Quinlan, 1986), naive Bayesian (Friedman et al., 1977; Mosteller and Wallace, 1964; Pearl, 1985), neural network (Lippmann, 1987), etc. A more recent approach is to explore strong relationships between specific sets of object features and their class labels; frequent patterns in records with the same class label can then be used to infer the class of other records with similar patterns. The important advantage in using AC over classical classification approaches is that the output of an AC algorithm is represented by simple *If-Then* rules which are easy and intuitive to interpret and understand.

Let S be a finite set of distinct class labels, each representing a suspect in our context. A training data set is a set of e-mails, each with an associated class label $S_i \in S$. A classifier is a function that maps an e-mail to a class $S_i \in S$. *AC* is the process of discovering *class association rules* (*CAR*) that capture the relationship between the combinations of stylometric features and the suspects. Specifically, the antecedent of a CAR contains a combination of stylometric features and the consequent of a CAR is a suspect. The support and confidence of a CAR have to pass the minimum support and minimum confidence thresholds specified by the operator. The notion of CAR is formally defined as follows.

Definition 2.2. (*Support of a rule*). *Let $A \rightarrow B$ be an association rule, where $A \subseteq V$, $B \in S$. The support of $A \rightarrow B$, denoted by $sup(A \rightarrow B)$, is the percentage of e-mails in E containing $A \cup B$.*

Definition 2.3. (*Confidence of a rule*). *Let $A \rightarrow B$ be an association rule, where $A \subseteq V$, $B \in S$. The confidence of $A \rightarrow B$, denoted by $conf(A \rightarrow B)$, is the percentage of e-mails containing B that also contain A.*

Definition 2.4. (*CAR*). *A class association rule has the form $A \rightarrow B$, where $A \subseteq V$, $B \in S$, $sup(A \rightarrow B) \geq min\_sup$, and $conf(A \rightarrow B) \geq min\_conf$, where min_sup and min_conf are the minimum support and minimum confidence thresholds specified by the user.*

For example, if 90% of suspect S_i's e-mails contain 3 paragraphs, then the confidence of rule $conf(3\ paragraphs \rightarrow S_i)$ is 90%. We can use this rule to classify future records that match this pattern. The minimum support threshold is used to avoid noise. Typically, AC finds the complete set of CARs that pass the user-supplied minimum support and confidence thresholds. When a new record requires classification, the classifier will select the matching rule with the highest confidence and support and use it to predict the class label. Recently proposed AC techniques will prune and rank rules and sometimes even use multiple rules to predict the class label of an unknown record as there are situations in which the single best rule may not be the most intuitive or even most appropriate choice. Many studies show that AC is intuitive, efficient, and effective.

Authorship attribution requires special attention when it comes to using AC techniques to obtain the best results; with multiple distinct classes and the need to consider much more than simply the strongest class, it becomes evident that a classifier should consider as much information as possible. Example 2.1 demonstrates why a single matching rule may not always be the best choice.

Example 2.1. *Suppose we want to find the author of an anonymous e-mail with feature items* (2, 5, 8). *The top* 3 *most confident rules matching the e-mail are as follows*:

Rule $R1$:2 → *Suspect* 0 (*support*:33%, *confidence*:90%)
Rule $R2$:5 → *Suspect* 1 (*support*:67%, *confidence*:89%)
Rule $R3$ 8 → *Suspect* 1 (*support*:50%, *confidence*:88%)

Most AC techniques that select the rule with the highest confidence would classify this e-mail as belonging to Suspect 0, but a closer look suggests that this decision has been made with no regard to the rest of the rule list. All three rules have similar confidence levels but both $R2$ and $R3$ have higher support which means that the values of those features were found more often in the training data set for Suspect 1. Suspect 1 is therefore a more intuitive choice and our algorithm should take this into account. Situations like this make it clear that in order to make a reliable and accurate prediction, especially when the result could mean the difference between guilty and innocent, an aggregate measure analysis based on multiple rules should lead to a more accurate classification.

Many studies have presented ways of greatly diminishing the quantity of class association rules so as to improve efficiency, given that usually only the strongest rule would be used for classification anyway. Our approach uses multiple rules (Li et al., 2001) and so it is important not to discard too much information. In general, rules with low support and confidence are pruned or outranked by more powerful rules, regardless of their class association. This means that a given author may be assigned to an unknown e-mail simply because he/she has a stronger writeprint than the true author and not based on a normalized measure of similarity. This would be the equivalent of identifying a matching fingerprint against two samples: one with a full print and another just with a partial print. The full print has more potential to match or to mismatch the unknown print, whereas the partial sample, even if it matches the unknown print very well, could still be discarded as its potential to fully match the unknown print is inherently lower. Once a set of *CAR*s is discovered, the next step is to remove common rules among the suspects because we only want the combinations of stylometric feature items that uniquely identify the author from a group of suspects.

After pruning common rules, the remaining list of CARs, denoted by *WPCAR*, encapsulates the writeprints of the suspects.

Definition 2.5. (*CAR Writeprint*). *The writeprint of a suspect S_i, denoted by $WP(S_i)$, is the set of rules in WPCAR with the form $A \rightarrow S_i$.*

Our proposed notion of CAR writeprint is different from the traditional authorship writeprint in previous works (Abbasi and Chen, 2008). The first distinction is that the feature item combination that composes the writeprint of a suspect S_i is generated dynamically based on the patterns present in their e-mails E_i. This flexibility allows us to concisely model the writeprint of different suspects by using various feature item combinations. Secondly, every rule in our notion of writeprint captures a writing pattern that can only be found in a single suspect's collection of e-mails. A cyber forensic investigator could then precisely point out a matched pattern in the malicious e-mail to support his/her conclusion of authorship identification. In contrast, a traditional classifier, such as decision tree, might use the same feature set to capture the writeprint of different suspects. It would be dangerous for the classifier to capture common writing patterns and use them as evidence that points to multiple authors; drawing a legal conclusion based on ambiguous evidence is problematic for obvious reasons. Our proposed notion of writeprint avoids this ambiguity and, therefore, produces more reliable and convincing evidence.

The removal of common patterns certainly improves the quality of the derived writeprint, especially for the purpose of evidence collection. However, one must understand the advantages as well as the disadvantages inherent in this technique. If there is a large number of suspects, it is entirely possible for one authors' writeprint to completely intersect with the union of the other authors' writeprints, leaving them without any writeprint at all. This could happen if the set of common rules is equivalent to the total set of rules for one class.

Refined problem statement

The problem of authorship attribution by multiple class association rules can be refined into three sub-problems:

1. to discover the CAR writeprint $WP(S_i)$ of each suspect S_i from the training e-mail sets E,
2. to identify the author of the malicious e-mail e by matching e with $WP(E_1), \ldots, WP(E_n)$, and
3. to extract clear and convincing evidence for supporting the conclusion on authorship.

These three sub-problems outline the challenges in a typical investigation procedure and reflect the use of AC in this process.

To solve sub-problems (1) and (2), we mine rules by extracting the frequent patterns and list of class association rules from the training set E while ranking and pruning them to build a representative final CAR list. For sub-problem (3), the matching group of rules with the best score serves as evidence for supporting the conclusion.

Classification by Multiple Association Rule for authorship analysis

In this section, we will present a novel data mining strategy that utilizes the concept of frequent stylometric patterns and AC to address the three authorship attribution sub-problems. Section 3.1 first presents a method that extracts frequent stylometric patterns and class association rules. Section 3.2 presents the procedure for pruning irrelevant or common rules that are shared among multiple suspects. Finally, Section 3.3 discusses how to use these rules to determine the most plausible author of a given anonymous e-mail.

Mining class association rules

A CAR list is compiled by mining a training data set to find the complete set of rules passing user-supplied minimum support and confidence thresholds. This is comparable to any frequent pattern mining or association rule mining task. *Classification by Multiple Association Rule* (*CMAR*) (Coenen et al., 2004; Li et al., 2001) forms the basis of the AC methods described in this study. The algorithm we use to mine rules is a variant of the *FP-growth* algorithm (Han et al., May 2000). Making use of efficient tree structures (Coenen et al., 2004), first a partial support tree and then a total support tree, database scans are minimized and there is no need to generate candidate sets. The benefits of this method are especially apparent when processing large data sets with a low support threshold and a large number of features; this situation is commonly seen in authorship attribution problems. Furthermore, accuracy is generally better when low support and confidence thresholds are used, making this choice of algorithm a suitable decision.

We illustrate the concept of class association rule mining below with examples. Refer to Li et al. (2001) for more details on the algorithm.

Suppose we have a set of features shown in Table 1. Setting the support threshold to 2 and confidence to 50%, the algorithm extracts class association rules as follows.

1) The feature set T is scanned, retrieving the set of feature items that pass the minimum support threshold. The set $F = \{a_1, b_1, c_2, d_3\}$ is called a frequent item set, as each element in the set appears at least twice. All other feature items appear only once and are pruned.

2) The feature items in F are then sorted in descending order to become $F = \{b_1, a_1, c_2, d_3\}$. The database is then scanned again to construct an FP-tree as shown in Fig. 1a. A FP-tree is a prefix tree with regard to the F-list. For each tuple in the feature set, feature items appearing in the F-list are extracted then sorted accordingly. For example, for the first tuple, (b_1, a_1) are extracted and inserted in the tree as the left-most branch. The author

Table 1
Feature set.

E-mail	Feat.A	Feat.B	Feat.C	Feat.D	Auth.ID
1	a2	b1	c2	d1	A
2	a1	b1	c2	d3	B
3	a3	b2	c1	d2	A
4	a1	b1	c3	d3	C
5	a1	b1	c2	d3	C

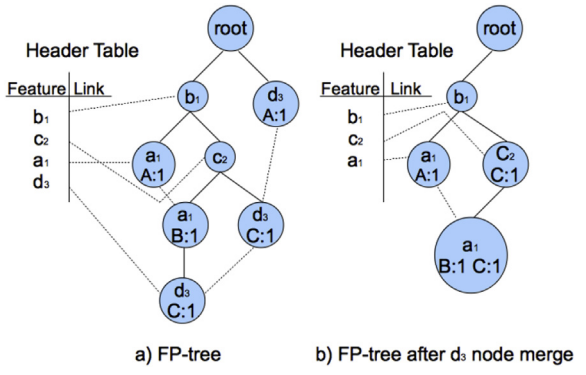

Fig. 1. Mining class association rules.

ID is appended to the last path node. For efficiency, feature set tuples always share prefixes. For example, the second tuple carries feature items (b_1, c_2, a_1) in the *F*-list and shares a common prefix b_1 with the first tuple. The sub-path with the left-most branch is therefore shared. All nodes sharing the same feature items are queued starting from the described header table.

3) The set of class association rules based on the *F*-List can be branched into 4 subsets with no overlap:
   - those with d_3
   - those with a_1 but not d_3
   - those with c_2 but not d_3 or a_1
   - those with b_1 exclusively

   These subsets are discovered iteratively one at a time.

4) Finding the subset of rules having d_3, the algorithm traverses nodes having feature item d_3 and looks upwards to construct a d_3-projected database, which contains three tuples: $\{b_1, c_2, a_1, d_3\} : C, \{b_1, c_2, d_3\} : C$ and $d_3 : A$. Given that all tuples containing d_3 are included, the challenge of finding all frequent patterns with d_3 in the entire feature set can be simplified to mining patterns in our d_3-projected database. Passing the support threshold, b_1 and c_2 are frequent feature items in the d_3-projected database. d_3 does not count as a local frequent attribute because in a d_3-projected database, d_3 is present in every tuple and therefore is trivially frequent. The projected database can be mined recursively by constructing FP-trees and other projected databases, as described in detail by Han et al. (May 2000). In our d_3-projected database, b_1 and c_2 always appear together, they are both sub-patterns of b_1c_2 and have the same support count as b_1c_2. The rule $R : b_1c_2d_3 \rightarrow C$ with support 2 and confidence 100% is generated based on author distribution. After processing all rules that include d_3, those nodes can be merged into their parent nodes. This means that any class association registered in any d_3 node is registered with its parent node, effectively shrinking the FP-Tree to what is shown in Fig. 1b. This operation is performed while the d_3-projected database is being built.

This process is then repeated for the remaining subsets of rules.

Pruning class association rules

Class association rule mining can generate an enormous number of rules; it is advantageous and rather simple to prune redundant or noisy rules in order to build a concise yet high quality classifier. The AC variant used in this study is modified from the rule ordering protocol called *CBA* (Liu et al., August 1998). The final rule list is ordered according to three ranking conditions. Given rules $R1$ and $R2$, $R1$ will be assigned a higher rank than $R2$, denoted by $R1 \succ R2$, if and only if.

1. $conf(R1) > conf(R2)$
2. $conf(R1) = conf(R2)$ but $sup(R1) > sup(R2)$, or
3. $conf(R1) = conf(R2)$ and $sup(R1) = sup(R2)$ but $|ant(R1)| < |t(R2)|$

Rule $R1 : P \rightarrow c$ is described as a general rule of rule $R2 : P' \rightarrow c'$, if and only if $P \subseteq P'$.

The first round of pruning uses ambiguous and high confidence rules to prune more specific and lower confidence rules. Given two rules $R1$ and $R2$, where $R1$ is a general rule with regard to $R2$. CMAR will prune $R2$ if $R1$ also has a higher rank than $R2$. The theory is that general rules with high confidence are more representative than more specific rules with low confidence, so we can prune the specific and low confidence ones. However, we will see that this may not be ideal behavior in an authorship attribution problem.

Rule $R1 : P \rightarrow c$ is said to be specific with regard to rule $R2 : P' \rightarrow c'$, if and only if $P \supseteq P'$.

While in general this pruning is harmless to accuracy and effective at making the process more efficient, one of our contributions is to prioritize more specific rules rather then more general ones. Therefore CMARAA, Classification by Multiple Association Rule for Authorship Attribution, orders rules slightly differently, changing condition 3) above to:

3) $conf(R1) = conf(R2)$ and $sup(R1) = sup(R2)$ but $|ant(R1)| > |ant(R2)|$

Part of the first round of pruning in CMARAA is therefore the opposite of what is done in CMAR. More specific rules with higher ranking are selected over more ambiguous rules. Intuitively the writing style patterns of an author should be as precise as possible in order to most accurately represent their more frequently occurring textual measurements. This change, in concert with other contributions that define CMARAA, allows the algorithm to achieve better results in terms of classification accuracy.

More general and more specific rule pruning is pursued when the rule is first inserted into the classification rule (CR) tree. When this happens, to check if the rule can be pruned or if other already inserted rules can be pruned we trigger retrieval over the tree.

The second round of pruning is done by only using rules that are positively correlated, determined by chi square testing. Only these positively correlated rules, i.e., those that pass user supplied a significance level threshold, are used during classification.

Chi square correlation based pruning is used to reflect only strong implications to perform classification. By removing rules that are not positively correlated, we reduce noise and make the classification process more efficient without negatively affecting accuracy. This pruning step is performed when a rule is being inserted into the CR-tree since the values necessary for performing the chi square test are readily available at this point.

The third pruning method builds a subset of high quality classification rules by performing database coverage pruning. A database coverage threshold Li (April 2001) is used to reduce the number of CAR's significantly, while maintaining the same representative number of rules per training record. This process is described in Algorithm 1.

Algorithm 1 Database coverage rule based selection.

Input: a list of rules and a database coverage threshold τ
Output: a concise but representative subset of class association rules
Protocol:

1. Order rules by rank;
2. For each training record, set the cover-count to zero;
3. For each rule R, find all matching training records. If rule R can appropriately classify at least one record, increase the cover-count of all records matching rule R by one. Remove a training record once its cover-count passes the database coverage threshold τ.

The database coverage method used by Li et al. (2001) is similar to the one used by Liu et al. (August 1998). The primary difference is that, instead of removing one data object from the training data set immediately after it is covered by some selected rule, it is left as part of the training set until such time that it is covered by a minimum of three other rules. The effect of this difference is that there will be more rules to consult when attributing a new object and therefore the unknown object will have a better chance of being classified accurately.

This pruning is performed once the rule mining process has been completed; it is the last pruning of rules described in CMAR.

One of this paper's contributions is the addition of another round of pruning for CMARAA. This last pruning method has been brought over from AM (Iqbal et al., 2010a) and is called common frequent item set elimination. When rules are being inserted into the CR-tree, any rule with the same antecedent as another distinct rule is flagged for removal. Once the CR-tree is processed, the flagged rules are removed. The reason that common rules are not removed immediately once discovered is that another rule for another author that is also common may also exist. It is therefore necessary to leave all rules in place until the process of generating all CAR's is complete.

Authorship classification

Once a set of rules is discovered and pruned for classification, as discussed in Sections 3.1 and 3.2, we are ready to classify anonymous e-mails. Given a test record, we collect the subset of matching rules from the CAR list. The rest of this section outlines how best to assign a class label based on the generated subset of rules.

If all rules that match the target object share the same class label, the test record is associated with that class label without contest.

If there exist two or more rules with different class labels, we create groups of rules for each class. All grouped rules share the same class label and each group is associated with a unique label. We then compare the strength of each group and associate the record with the strongest one.

To appropriately compare the groups' strengths, we must measure the combined effect of each group. Intuitively, if the group's rules are highly positively correlated and have a relatively high support metric, the group's effect should be strong.

Typical AC algorithms use the strongest rule as a representative, which means that the single rule with the highest rank is selected. The danger of simply choosing the rule with the highest rank is that this may favor minority classes, as illustrated by Example 3.1.

Example 3.1. . *In an authorship attribution exercise, there are two rules*:

R1. *Feature A* = no → *Author B*(*support* = 450, *confidence* = 60%)

R2. *Feature B* = *yes* → *Author A*(*support* = 200, *confidence* = 99:5%)

See observed and expected values for rules *R1* and *R2* in Table 2.

Based on the measured and expected values, the chi square value is 97.6 for *R1* and 36.5 for *R2*. For an anonymous e-mail with no feature A and feature B, we may

Table 2
Observed and expected values.

	Author A	Author B	Total
(a) R1 Observed			
R1			
Feature A	410	40	450
No Feature A	20	30	50
Total	430	70	500
(b) R2 Observed			
R2			
Feature B	209	1	210
No Feature B	241	49	290
Total	450	50	500
(c) R1 Expected			
R1			
Feature A	387	63	450
No Feature A	43	7	50
Total	430	70	500
(d) R2 Expected			
R2			
Feature B	189	21	210
No Feature B	261	29	290
Total	450	50	500

predict that the author would be Author B rule $R1$, if the choice between rules is based only on chi square values. However, rule $R2$ is clearly much better than rule $R1$ since rule $R2$ has much higher confidence. This presents a challenge in determining which rule is the strongest.

Using the compound of correlation of rules as a measure is one alternative. For example, we can sum up the values in a group as the strength measure of the group, but this would suffer from the same problem that it may favor minority classes.

A better way would be to integrate both correlation and popularity into the group measure and so we have adopted a weighted measure Li (April 2001) called Weighted Chi Square (WCS). For each rule $R: P \rightarrow c$, let $sup(c)$ be the number of records in the training data set that are associated with class label c and let $|T|$ be the number of data records in the entire training data set. Equation (1) defines the max chi square value, used as the upper bound of the chi square value of the rule.

$$max\chi^2 = \left(min(sup(P), sup(c)) - \frac{sup(P)sup(c)}{|T|}\right)^2 \left|T\right| e \quad (1)$$

where

$$e = \frac{1}{sup(P)sup(c)} + \frac{1}{sup(P)(|T| - sup(c))} + \frac{1}{(|T| - sup(P)sup(c))} + \frac{1}{(|T| - sup(P))(|T| - sup(c))}$$

For each group of rules, the weighted measure of the group is calculated using Equation (2).

$$\sum \frac{(\chi^2)^2}{max\chi^2} \quad (2)$$

As demonstrated, the ratio of the χ^2 value against its upper bound, max chi square, is used to overcome the bias of the chi square value favoring any minority class. Li et al. (2001) noted that it was difficult to theoretically prove the soundness of measures on strength of rule groups. Instead, they tested and assessed the effect of measures through empirical observation and according to their experimental results, WCS got the best results of a good set of other candidates.

Finally, CMARAA's third contribution is the intuitive output of the class association rule groups. Let A be the group of suspected authors and let F_i be a feature in a feature set F comprising of hundreds of potential features used by the algorithm. A group of classification rule defined by three features strongly associated with a class would look like this:

$$F_1 + F_6 + F_{90} \rightarrow A_2$$

$$F_5 + F_{124} \rightarrow A_2$$

$$F_{45} + F_{89} + F_{94} + F_{213} \rightarrow A_2$$

Experimental evaluation

To evaluate the accuracy, efficiency and scalability of the Classification by Multiple Association Rule (CMAR) algorithm and our proposed augmented implementation of it, CMAR for Authorship Attribution (CMARAA), we have compiled results from a comprehensive performance study. In this section, we compare CMAR and CMARAA, against two well known classification methods: Classification by Association (CBA) Liu et al. (August 1998) for comparison against a baseline AC algorithm and AuthorMiner (Iqbal et al., 2010a) (AM), the previous leader in data mining based authorship attribution. In addition, CMARAA is evaluated against some well-known classifiers that are commonly used in most authorship studies including Naive Bayes, Bayesian Networks (BayesNet), Ensemble of Nested Dichotomies (END), and Decision Trees (e.g., J48). It shows that CMARAA matches or outperforms the other methods including CBA (Liu et al., August 1998) and AM (Iqbal et al., 2010a) in terms of average accuracy. Other classification techniques like Random Forest, SMO and SVM were not included as they are more difficult to interpret for users and one of the goals of this study is to generate intuitive output.

All tests have been performed on a 3.4 GHz Core i7 with 12G main memory, running Mac OS 10.7.3. CMAR (Li et al., 2001) and CBA (Liu et al., August 1998), were implemented by Frans Coenen, in the course of demonstrating the power and scalability of their Apriori-TFP method (Coenen et al., 2004). AuthorMiner (Iqbal et al., 2010a) was implemented by its authors.

The e-mail collections used in this study are all from the Enron e-mail data set[3] which is the most suitable collection of publicly available e-mails according to B. Allison and L. Guthrie in their paper Allison and Guthrie (2008). Specifically, we have selected hundreds of e-mails from authors Paul Allen, John Arnold, Sally Beck, John Dasovich, Mark Headicke, Vince Kaminsky, Steven Kean, Kam Keiser, Philip Love, Kay Mann, Susan Scott, Carol St Clair, Kate Symes and Kim Watson. Although there were many other authors, the ones listed above had the largest collections. E-mail headers and all content not written by the respective authors for the specific message have been removed. The purpose of this cleaning was to ensure that each e-mail consisted of text written solely by its author. In general only about 20% of e-mails in the original data set were retained because the rest consisted of text shorter than a few sentences, forwarded materials or attachments. Similarly for test objects, it would not be prudent to expect the accurate classification of any message consisting of a few words or a single sentence. Messages that short would be trivial to alter in such a way that they contain no specific writing style whatsoever, or even a spoofed writing style if the malicious entity was well versed in authorship attribution methodologies.

All results are accuracy averages compiled from running the various algorithms on data sets with the same number of authors but each containing a different combination of authors, in order to show that results are stable and not simply the result of lucky guesses or hand picked sets that support our conclusions. As to be expected, sets of authors with low accuracies show lower accuracies across the board and vice-versa. Each test is fair and the data sets have not

[3] http://www.cs.cmu.edu/~enron/.

been altered in any way to give advantage to any one algorithm over the others.

All figures and results consider the percentage of correctly matched authors in the testing set to reflect accuracy of authorship identification. For example, if there are 4 test records and 3 of them are correctly matched to an author, the accuracy will be 75%. The tests have been performed by splitting the entire data set into training and testing partitions with a ratio of 90% training data to 10% testing data. This means that if there are 1000 records in the entire data set, 900 records will be used for training and 100 for testing. For AuthorMiner and the AC algorithms, the training sets are used to discover frequent patterns for the classifier and then each e-mail in the test set is classified and verified. The training and test set splits are done on a per author basis so each author's data set is split by the user-supplied percentage and the respective author sets are combined into a global training set and testing set. This separation is done using the same method for all tested algorithms in order for results to be directly comparable.

In order for tests to be repeatable, the training and testing set split is done in order and not at random, with the first portion belonging to the training set and the rest to the test set. E-mails are named numerically and are input in ascending alphabetical order. This does potentially cause the results to be skewed towards how easy or hard it is to classify e-mails found at the end of the set, but each algorithm must deal with this issue, again ensuring that results are comparable and fair to the strengths of each algorithm. The rest of this section describes the figures showing the results of the various tests.

For the experiments, the parameters are set as follows.

For CBA and CMAR, we set the support threshold to 10% and the confidence threshold to 0%. The reason the confidence threshold has been set to 0% is to demonstrate the effect of the final round of pruning in CMARAA has on accuracy. A high confidence threshold would effectively eliminate common rules among multiple authors as explained previously. Furthermore, setting a different confidence threshold for the various algorithms would make fair comparison impossible. Given that the confidence threshold in CMAR, CBA and CMARAA does not affect accuracy, it was deemed safe and fair to simply set it to 0%. Note that the confidence and support thresholds are set mainly to improve efficiency, not accuracy.

CMARAA uses the same support and confidence thresholds as CBA and CMAR but the support threshold is considered on a per author basis instead of applying it to the size of the entire training data set. This allows for better extraction of frequent patterns on a per author basis. This is important because without this distinction, CMARAA would technically be looking for frequent patterns across all authors instead of patterns that are representative for a single author. Without this contribution, it would be harder to associate anonymous e-mails with their author, as the results demonstrate.

Figs. 2 and 3 show average classification accuracies over sets of collections of e-mails for 2 to 10 authors for each algorithm tested in this study. Accuracies range from 30% to 92% with the most accurate algorithms being CMAR for Authorship Attribution (CMARAA), Classification by Association (CBA), Classification by Multiple Association Rule (CMAR), and AuthorMiner (AM), respectively. The accuracies decrease as the number of authors increases, with CMAR seeing the biggest drop when there are 10 authors. The classification result depends more on how unique the writing styles of the included authors are than the strength of the algorithms; naturally, two authors who have similar writing styles will be harder to tell apart, regardless of which technique is employed.

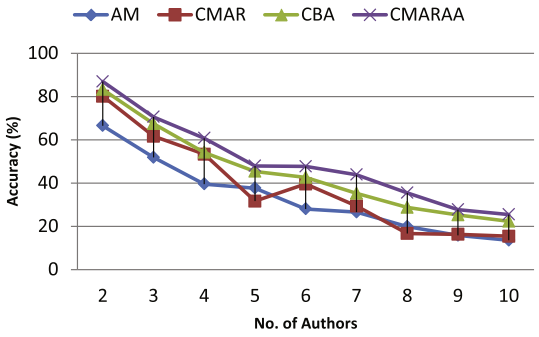

Fig. 2. Accuracy vs. Number of authors.

Authors	AM	CMAR	CBA	CMARAA
2	66.489	80.106	83.229	86.923
3	51.83	61.63	67.37	70.54
4	39.52	53.33	54.12	60.73
5	37.602	31.636	45.34	47.9
6	28	39.535	42.555	47.6925
7	26.5275	29.37	35.2075	43.8675
8	19.8425	16.6633333	28.705	35.44
10	15.78	16.32	25.2166667	27.6966667

Fig. 3. Accuracy vs. Number of authors values.

With the minimum support threshold held constant, CMAR no longer generates rules with every run when there are many authors; this is due to the fact that the writing styles of authors are usually distinct from one another and so the support of feature items will not always pass the minimum support threshold, which is a percentage of records across the entire training set. For example, if the support threshold is 10% and there are 10 authors with 100 e-mails each, then a feature item unique to one author would need to appear in every single e-mail of that author in order to be considered frequent. One of the main contributions of this study, implemented in CMARAA, reduces the support threshold to consider only one author's training set, allowing feature items that are frequent, even if unique to one author, to be considered.

Finally, the proposed method is evaluated against some common authorship attribution classifiers. Figs. 4 and 5 depict experimental average results from AuthorMiner, Naive Bayes, Bayesian Networks (BayesNet), Classification by Multiple Association Rule (CMAR), Classification by Association (CBA), Ensemble of Nested Dichotomies (END), Decision Trees (e.g., J48) and CMARAA. The test was to differentiate between 2 authors over 6 runs of distinct author pairs. Showing the same trend of Fig. 1, CMARAA is better than or equivalent to all other methods.

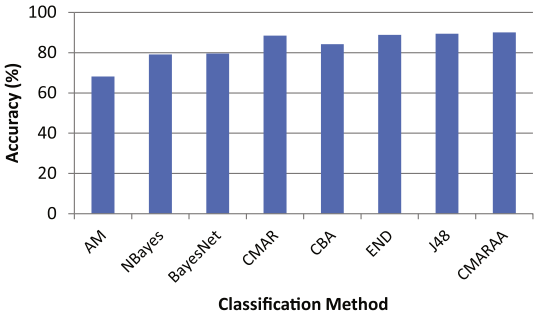

Fig. 4. Accuracy vs. Classification method.

Dataset	AM	NBayes	BayesNet	CMAR	CBA	END	J48	CMARAA
2a	65	75	75	100	100	95	95	100
2b	57	85	85	80.18	92.79	91.96	91.96	83.78
2c	75	76	76.47	87.5	87.5	79.41	79.41	87.5
2d	66	76.31	76.31	81.89	81.08	89.47	89.47	83.78
2e	64.86	78.51	80.74	83.33	86.11	88.1	91.85	91.66
2f	81.25	83.64	83.85	97.92	57.58	89	89.02	93.75
Average	68.19	79.08	79.56	88.47	84.18	88.82	89.45	90.08

Fig. 5. Accuracy vs. Classification method values.

We do not provide extensive CPU or I/O metrics in this study. In the context of a criminal investigation, we assume that execution time is of little importance compared to the quality of classification and evidence collection. With that said, Fig. 6 shows the average number of seconds each algorithm took to complete the classification process on sets of 2, 3, 4 and 5 authors. The algorithms that required the least CPU time for 2 authors were CBA, CMAR, CMARAA and AM respectively. For 3 authors, CMAR came in first, with CBA, CMARAA and AM in second, third, fourth and fifth place respectively. The same order of fastest times held true for 4 authors. For 5 authors, CBA came in first place followed by CMAR, then CMARAA and finally AuthorMiner.

The run times presented in Fig. 6, are primarily the result of their respective frequent item set discovery processes. AuthorMiner uses the original Apriori (Agrawal et al., 1993) algorithm for discovering frequent item sets, whereas CMAR, CBA and CMARAA all use a much faster variant of the FP-Growth (Han et al., May 2000) algorithm, using more efficient tree structures to represent frequent item sets and class association rules. Technically AM could use the FP-growth (Han et al., May 2000) method, but improving its performance was not a goal of this study.

These results demonstrate reliable and repeatable proof that authorship attribution data mining methods can be very useful. This also proves, once again, that the writing styles of authors can be modeled by extracting patterns from transformed semantic content to create individually recognizable writeprints.

Number of Authors	Average Algorithm Run Time (Seconds)			
	AM	CMAR	CBA	CMARAA
2	15.717	3.132	3.0599	8.1525
3	52.83	7.27	7.83	55.39
4	575.74	8.44	8.51	96.34
5	443.467	13.24	12.9844	108.7104

Fig. 6. Algorithm run time.

Conclusion

Cybercrime investigations are in need of a state-of-the-art computer aided writeprint modeling algorithm that can provide reliable evidence to support authorship attribution. When someone's innocence or guilt is on the line, it is very important to have access to the most accurate and efficient methods of linking suspects to the evidence collected.

This study explores the application of a promising data mining technique called associative classification on the e-mail authorship attribution problem for the first time. Additionally, we propose that class-based AC allows for the extraction of a more accurate and easier to understand writeprint. We also acknowledge that modifying the rule pruning and ranking system described in the popular Classification by Multiple Association Rule (CMAR) algorithm to prioritize more specific patterns can also provide a more accurate writeprint. The removal of patterns common among various authors that results in a relatively unique writeprint and the easy to understand output makes for more convincing evidence in a court of law. The presented customized AC method helps fight cybercrime by addressing the e-mail authorship attribution problem. We do not claim that the findings of our proposed method will be enough to convict a suspect on its own, but it can certainly complement other evidence and therefore positively influence the court's decision.

Acknowledgments

The authors would like to thank the reviewers and the editor for the thorough reviews and valuable comments, which significantly improve the quality of this article. This research is supported by the Research Incentive Fund (RIF) #R13059 from Zayed University and Natural Sciences and Engineering Research Council of Canada.

References

Abbasi A, Chen H. Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Trans Inf Syst 2008;26(2):1–29.

Agrawal R, Srikant R. Mining sequential patterns. In: Proc. of the 11th International Conference on Data Engineering (ICDE); 1995. p. 3–14.

Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD International Conference on Management of Data; 1993. p. 207–16.

Allison B, Guthrie L. Authorship attribution of e-mail: comparing classifiers over a new corpus for evaluation. Bente Maegaard Joseph Mariani Jan Odijk Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair). In: Choukri Khalid, editor. Proceedings of the sixth International Conference on Language Resources and Evaluation (LREC'08); 2008. Marrakech, Morocco, may 2008. European Language Resources Association (ELRA).

Coenen F, Goulbourne G, Leng P. Tree structures for mining association rules. Data Min Knowl Discov 2004;8:25–51.

Corney M, de Vel O, Anderson A, Mohay G. Gender-preferential text mining of e-mail discourse. In: Proc. Of the 18th Annual Computer Security Applications Conference (ACSAC); 2002. p. 282.

de Vel O. Mining e-mail authorship. KDD August 2000.

de Vel O, Anderson A, Corney M, Mohay G. Multi-topic e-mail authorship attribution forensics. In: Proc. Of ACM Conference on Computer Security – workshop on data mining for security applications; 2001.
de Vel O, Anderson A, Corney M, Mohay G. Mining e-mail content for author identification forensics. SIGMOD Rec 2001b;30(4):55–64.
Ding SHH, Fung BCM, Debbabi M. A visualizable evidence-driven approach for authorship attribution. ACM Trans Inf Syst Secur (TISSEC) March 2015;17(3):12:1–12:30. article no: 12.
Fachkha C, Bou-Harb E, Boukhtouta A, Dinh S, Iqbal F, Debbabi M. Investigating the dark cyberspace: profiling, threat-based analysis and correlation. In: Proceedings of the 2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS), CRISIS 12. Washington, DC, USA: IEEE Computer Society; 2012. p. 1–8.
Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn 1977;29:131–63.
Grieve J. Quantitative authorship attribution: an evaluation of techniques. Lit Linguist Comput July 2007;22(3).
Han J, Yin X. Cpar: classification based on predictive association rules. In: Proc. of the third society for industrial and applied mathematics. society for industrial and applied mathematics; 2003.
Han J, Kamber M, Pei J. Data mining: concepts and techniques (The Morgan Kaufmann series in data management systems). 2006. Morgan Kaufmann, 2nd edition, January 2006.
Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. SIGMOD'00 May 2000:1–12.
Iqbal F, Binsalleeh H, Fung BCM, Debbabi M. Mining writeprints from anonymous e-mails for forensic investigation. Digit Investig 2010:1–9.
Iqbal F, Khan LA, Fung BCM, Debbabi M. E-mail authorship verification for forensic investigation. In: Proc. of the 25th ACM SIGAPP symposium on applied computing (SAC). Sierre, Switzerland: ACM Press; March 2010b. p. 1591–8.
Iqbal F, Binsalleeh H, Fung BCM, Debbabi M. A unified data mining solution for authorship analysis in anonymous textual communications. Inf Sci Special Issue Data Min Inf Secur May 2013;231:98–112.
Moshe Koppel and Jonathan Schler. Authorship verification as a one-class classification problem.
Ledger GR, Merriam TVN. Shakespeare, Fletcher, and the two Noble Kinsmen. Lit Linguist Comput 1994;9:235–48.
Li W. Classification based on multiple association rules. April 2001.
Li W, Han J, Pei J. CMAR: accurate and efficient classification based on multiple class-association rules. In: In proc. of ICDM; 2001.
Lippmann RP. An introduction to computing with neural networks. IEEE Acoust Speech Signal Process Mag 1987;4(2):4–22.
Liu B, Hsu W, Ma Y. Integrating classification and association rule mining. KDD August 1998:80–6.
Mendenhall TC. The characteristic curves of composition. Science 1887;11(11):237–49.
Mosteller F, Wallace DL. Applied Bayesian and classical inference: the case of the Federalist papers. 2nd ed. New York: Springer-Verlag; 1964.
Pearl J. Bayesian networks: a model of self-activated memory for evidential reasoning. In: Proc. of the 7th Conference of the cognitive science society; 1985. p. 329–34.
Quinlan JR. Induction of decision trees. Mach Learn 1986;1(1):81–106.
Stamatatos E. A survey of modern authorship attribution methods. J Am Soc Inf Sci Technol (JASIST) March 2009;60:538–56.
Teng G, Lai M, Ma J, Li Y. E-mail authorship mining based on svm for computer forensic. In: Proc. of the 3rd International Conference on Machine Learning and Cyhemetics; August 2004.
Thabtah F, Cowling P, Peng Y. Mcar: multi-class classification based on association rule. In: ACS/IEEE 2005 International Conference on Computer Systems and Applications; 2005. p. 33. aiccsa.
Zheng R, Qin Y, Huang Z, Chen H. Authorship analysis in cybercrime investigation. In: Proc. of the NSF/NIJ symposium on intelligence and security informatics (ISI); 2003. p. 59–73.