An algorithm for automated authorship attribution using neural networks

Matt Tearle, Kye Taylor and Howard Demuth
University of Colorado, Boulder, CO, USA

Abstract

We present an algorithm as evidence of the possibility of a truly automated stylometric authorship attribution tool, based on committees of artificial neural networks. Neural networks have an advantage over traditional statistical stylometry in that they are inherently nonlinear, and therefore can consider nonlinear interactions between stylometric variables. The algorithm presented (1) is intended to demonstrate the feasibility of an automated approach using neural networks and (2) highlights important areas for further research. We present results of two separate test experiments—Shakespeare and Marlowe, and the Federalist Papers—as a demonstration of the method's generality. In both cases, our algorithm produces committees that correctly predict the test works, without requiring the usual precursory statistical study to determine efficacious stylometric measures.

Correspondence:
Matt Tearle, The MathWorks, 3 Apple Hill Dr, Natick, MA 01760, USA.
E-mail: mtearle@gmail.com

1 Introduction

Although, determining the authorship of a written work is an old problem (historically limited to 'academic' questions such as whether a certain person was the true author of Shakespeare's works) it is just as important today, with applications to criminal investigation, security, and questions of academic integrity. This is particularly true with the advent of email and other electronic means of communication with which an author's identity can be masked (de Vel *et al*., 2001; Zheng *et al*., 2003). Indeed, the prevalence of electronic written copy means that author identification, in the case of dispute, must be determined entirely from the text itself, since forensic investigation of other factors such as ink or handwriting is nonexistent. This leaves two possible approaches: investigation of either the content or the form of the text. The former is the domain of traditional literary analysis; the latter is the domain of *stylometry* or *literary stylistics*: attempting to quantify an author's 'style' by studying objective measures of the text.

The list of textual metrics that could be considered is, technically, endless; in reality, we wish to consider only those metrics that have some demonstrable amount of discriminatory power. Many different metrics have been considered by various investigators (Damerau, 1975; Merriam, 1989; Martindale and McKenzie, 1995; Merriam, 2003; Mannion and Dixon, 2004) with the degree to which each metric can distinguish reliably between different authors evaluated for various specific cases. Not surprisingly, each metric works to a different level of reliability for different authors—for example, sentence length alone may distinguish Oliver Goldsmith from other authors (Mannion and Dixon, 2004), but it is typically considered a poor discriminator in general (Holmes, 1994). Consequently, some statistical metrics and some determinations (using a stylometric approach) of specific authorial debates have been criticized (Valenza, 1991;

Literary and Linguistic Computing, Vol. 23, No. 4, 2008.
doi:10.1093/llc/fqn022 Advance Access Published on 22 October 2008

Elliott and Valenza, 1996; Foster, 1996), leading to questions being raised as to the usefulness of the whole statistical approach to author identification (Milic, 1991; Potter, 1991). Part of this criticism can be traced to the lack of any success in applying stylometry in an automated manner: each new authorial question requires a decision as to which metrics to consider and which have discriminatory power. Typically this decision alone requires extensive effort, calculating all the metrics from the text and performing statistical analysis of them. Furthermore, some amount of testing is required to then demonstrate that the stylometric process thus derived also generalizes to other works, before then being applied to the unknown work in question. It is, therefore, hardly surprising that this approach appears to offer, at best, only minimal advantage over the traditional approach of studying textual stylistics. Nonetheless, a number of metrics have been shown to be useful discriminators, at least to some extent, across a wide variety of authors (Holmes, 1994). The problem, then, becomes one of choosing some number of the best metrics from this list of likely candidates.

One recent attempt to automate the process of determining authorship using a given set of stylometric statistics has been the use of *artificial neural networks* (ANNs) (Singh and Tweedie, 1995). An ANN is a mathematical construct, invariably implemented on a computer, that is traditionally viewed as a very simple model of a far more complicated natural neural network: the human brain. Humans empirically determine rules about the world around them by observing many examples of cause-and-effect and then finding patterns in these data. Similarly, an ANN can be trained to fit the pattern linking a set of input data to a corresponding set of outputs or targets. Therefore, an ANN may be designed and trained to mimic the behavior of a human literary expert, who has learned to distinguish authors by reading many examples of their known works. Questions about unknown works, then, may be answered by this expert (human or artificial), based on accumulated knowledge extrapolated to the unattributed work. This approach has been applied successfully to works of Shakespeare and his contemporaries (Matthews and Merriam, 1993; Merriam and Matthews, 1994; Lowe and Matthews, 1995) and to the case of the *Federalist* papers (Tweedie *et al.*, 1996). However, in order to train and use an ANN, a choice must be made as to what data should be given as inputs. We are, therefore, faced with the same problem as with any stylometric analysis: choosing an appropriate set of input metrics.

Despite the success of the use of ANNs in the specific problems mentioned above, there has not been any sudden proliferation of neural computing-based authorship studies. Neither has there been significant discussion in the literary computing literature of how this approach can be made into a useful, generally applicable tool that will promote widespread usage among the literary analysis community. In this article, we present an algorithm and some results from it that demonstrate the possibility of a truly automated procedure. As has been noted by many researchers, no one technique will provide conclusive proof of authorship, but each different technique can contribute to the overall conclusion. We therefore believe that a general, easily applicable stylometric tool would still be a useful contribution to the field. Naturally, the algorithm presented here does not yet completely achieve this goal: in particular, it is still computationally intensive; it does, however, demonstrate the possibility of such a method, and also, importantly, highlights the areas that currently need research in order to fully realize this goal.

2 Background

2.1 ANNs

An ANN is designed to make a prediction by generalizing known examples, which are used to *train* the network. Mathematically, the network must transform a set of inputs into a set of outputs. The inputs and outputs are numbers, so we can represent a set of m inputs or outputs as a vector: $\mathbf{x} = [x_1, x_2, \ldots, x_m]$. In order to train the network, we need some number n of input vectors $\{\mathbf{x}_k\}_{k=1}^{n}$ with known corresponding outputs $\{\mathbf{y}_k\}_{k=1}^{n}$.

An ANN can be described as a number of *neurons*, arranged in *layers*, connected to each other

with *weights*, via *transfer functions* (Hagan *et al.*, 1995). The arrangement of the neurons determines the number of weights and their effect on the final output. For a classification problem, the network would have m neurons in the *input layer* (one for each of the m elements of the input vector $\mathbf{x}$), some number, n_h, of neurons in a *hidden layer*, and a single *output-layer* neuron that, ideally, would take only one of two values, representing a decision between two possibilites. With a weight for each connection between neurons, this gives a total of $n_h(m+1)$ weights. The values of the weights, which are seen as analogous to the different strengths of the neural connections in the brain, determine how the network will transform a set of inputs into an output. Mathematically this is nothing more than choosing a complicated function $\mathbf{f}(\mathbf{x})$ that maps the inputs to the outputs. This function has a large number of parameters (the weights) that may be adjusted; the choice of network architecture (the number and arrangement of neurons, and the type of transfer functions) determines the form of this function. Training the network is a matter of adjusting the weights so that the network correctly returns observed targets, $\{\mathbf{y}_k\}_{k=1}^n$, from the observed inputs, $\{\mathbf{x}_k\}_{k=1}^n$, analogous to the brain adjusting the strength of neural connections after repeated observation (e.g. repeatedly hearing the sound 'cat' while being shown a picture of a cat produces a strong neural connection between different parts of the brain, producing a mental connection between the word and the object it symbolizes). Again from a purely mathematical viewpoint, this is the problem of choosing the parameters of the function so as to minimize the error in the result given by the function—that is, the difference between the output of the function for a given set of inputs, $\mathbf{f}(\mathbf{x}_k)$, and the actual output corresponding to those inputs, $\mathbf{y}_k$.

2.2 Training

Finding the weights so that the error is minimized is a standard mathematical problem: minimization of a function (the error) of many variables (the $n_h(m+1)$ weight values). The error function value is typically taken to be the sum of the squares of the individual errors: $E = \sum_{k=1}^{n} \left(\mathbf{f}(\mathbf{x_k}) - \mathbf{y_k}\right)^2$. Ideally, we wish to find the weights that give the smallest value of E over all possible sets of weights. In practice, such a global minimization problem is infeasible; a practical alternative is to find any local minimum for which the resulting value of E is smaller than a chosen small tolerance. In this way, we ensure that we have the best set of weights within some limited range, and that the resulting network is accurate within a given tolerance on the training data—i.e. even if we do not have the best possible weights, we know they are good enough to give a functional network.

To find a local minimum in the error function, we may use any standard mathematical algorithm for multidimensional minimization, such as conjugate gradient or steepest descent. For the work shown here, we use the Levenberg–Marquardt method—a common choice for ANN applications (Hagan *et al.*, 1995). These algorithms work iteratively, starting with an initial guess—typically random—for the weights and repeatedly improving the values until a local minimum is found. The local minimum found is dependent on the initial values, with some initial guesses leading to viable networks and others leading either to a local minimum with an unacceptably high error or to the minimization method not converging to a solution at all.

2.3 Architectural issues

Ideally, the weights would be chosen to produce no error; in reality, however, it is likely that there will be random noise in the training data so that it is not possible to give zero error without requiring the function $\mathbf{f}$ to be overly complicated. For example, a set of 100 (x, y) data pairs may be experimentally collected, with y linearly related to x, but it is probable that only very few of the 100 points would lie exactly on the theoretical line relating the variables. It would, therefore, be impossible to fit a line to the data with zero error. It would, however, be perfectly possible to fit a ninety-ninth degree polynomial to the data that would have zero error. This polynomial fit, although it would give perfect results on the training data, would be utterly useless for predictive purposes, since it would oscillate wildly between the data points, as opposed to a straight line fitted through the points, giving good, but

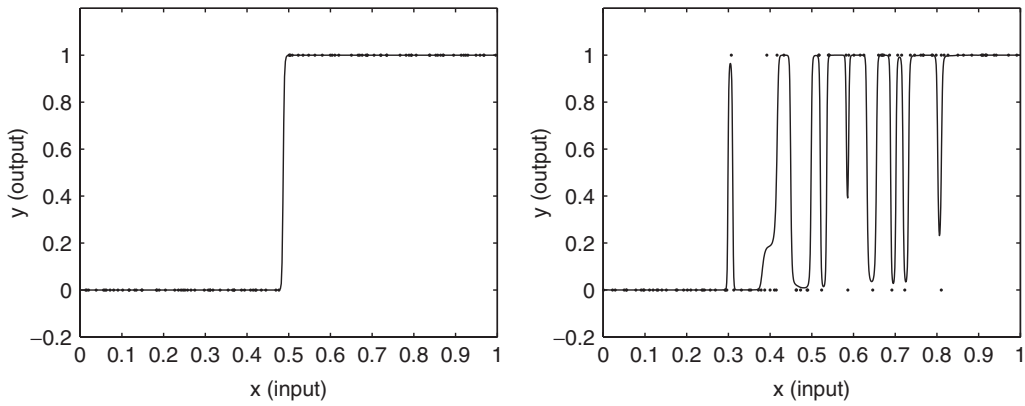

Fig. 1 The difficulties with binary data with noise. In this simple example, the output is determined by a single deciding factor: if $x < 0.5$ then y is one type ('0') and if $x > 0.5$ then y is the other type ('1')

not perfect, results. Since the entire purpose of using an ANN is to *generalize* from known examples, we wish to avoid this *overfitting* phenomenon. On the other hand, a very simple function will not capture all the behavior of the data, leading to a higher error in the fit of the model to the measured data. This is particularly problematic for the application to authorship classification since (1) there is likely to be significant noise in the data, due to the statistical nature of the underlying model (that is, even though one author tends to use longer sentences than another in general, in any given specific sample it is entirely possible for the opposite to be true) and (2) the output is binary, rather than a continuum—i.e. since our goal is classification into 'Author A' or 'Author B', mathematically we are fitting a function to outputs of either 0 or 1 (or −1 and 1, or any other two discrete values). These combined difficulties are illustrated in Fig. 1, in which the fitting function determined by a simple network is shown. In the case of the idealized data, a network with only one hidden-layer neuron (and, therefore, only two weights) can fit the data to within a mean-squared error of $<10^{-8}$; in the case of the noisy data, we could do no better than an error of 0.02 which still required fifty hidden-layer neurons (100 weights) and produces a model that, predictably, overfits the data. Of course, this example oversimplifies the situation: in reality, there are several input variables. Consequently, the data points live in a high-dimensional space, and the 'noisy' points are not so obviously 'out of place'. Nonetheless, the ultimate goal is still to find a function **f** that successfully classifies all the points in as simple a manner as possible (such as '$x < 0.5$ implies $y = 1$').

Since the complexity of the function **f** depends on the number of weights, and, consequently, the number of neurons, we find ourselves trying to steer between the Scylla of too few neurons (leading to a network that cannot be successfully trained) and the Charybdis of too many (leading to a network that trains easily but is predictively useless). In terms of the analogy with human experts, the former situation is like a dullard unable to see any difference between the canonical works provided, and the latter is like a pedantic savant who finds any manner of spurious correlations in the canonical works but cannot therefore provide any insight to new works.

Complicating the issue further is the question of how much data should be used for training purposes. In many applications, a decrease in predictive power is observed when a large number of training data is used to train the network. Although seemingly counter-intuitive, again this is the mathematically natural result of trying to force a fitted function too close to too many points (as in Fig. 1). The specific problem of author identification, however, has an intrinsic limitation which, in fact, reduces the probability of overtraining: the

available training data is necessarily limited. In many ANN applications, a limitless set of training data can be created; we, however, cannot ask Christopher Marlowe to pen a few more plays so that we have more available data. We therefore do not consider the issue of limiting the training data, and concentrate instead on the question of network configuration.

Determining the optimal network architecture for a given application is a difficult problem and an area of ongoing research in the ANN community. This is one of the key issues for our eventual goal of creating a fully automated mechanism useful to the literary computing community, and should be considered an important area for further research.

Finally, it should be noted that whether or not a 'good' network even exists does depend on the data. Considering the simple example in Fig. 1, we see that the network we would hope for (in the left-hand graph) is probably not even possible in the case of noisy data (right-hand graph). To avoid Scylla, we must have a large number of neurons, forcing us irresistibly into the poorly generalizing Charybdis shown in the right-hand graph. The impossibility of steering a middle course shows us that the data set itself is inadequate; because of the statistical nature of the data—and the noise therein—a single input variable does not contain enough discriminatory power to determine the output perfectly. It is necessary to include more inputs so that the input data are more separated in the higher dimensional space. However, including more inputs necessarily increases the complexity of the function **f**. Therefore, choosing the correct set of inputs is an important step in the creation of the network; we will consider this aspect in greater detail in Section 2.6.

2.4 Validation

In order to avoid the problem of over-generalization discussed above, it is common to perform some form of *cross-validation*: some of the canonical works are set aside (at the necessary expense of depleting the training data available) and the successfully trained network is then applied to these validation works as a test of its ability to generalize. Such techniques, of varying sophistication, have been used in previous studies on attribution with ANNs; we consider only the simplest approach of applying each validation work individually and measuring the network's performance. Given its critical nature to developing an automated approach, it is important to consider how validation performance can be quantified. In a perfect world, the network would return a value of 0 or 1 for any given work; in reality, it will return a value somewhere between these two extremes. (Depending on the choice of transfer function between the hidden and output layers, it is possible to create a network capable of returning values outside this range. We choose to use a *log-sigmoid* transfer function, thus forcing the output to lie between 0 and 1.) Therefore, we already have ambiguity as to how to judge validation performance: do we consider the actual values or simply the decision (rounding to 0 or 1)? And if the latter, do we consider everything <0.5 to be a vote for Author 0, or do trisect the interval to allow for a region of ambiguity? By analogy with a human expert, we may view the network's return value as indicating some degree of certainty. For example, do we have more confidence in one network that classifies ten works of Author 0 all as 0.4 than another network that classifies eight of them as 0 and 2 as 1, or another that classifies 6 as 0 and 4 as 0.6? In the first case, the expert has a 100% success rate but is 'unsure' of every decision; in the second case, the expert is certain on all decisions but is wrong 20% of the time; the third expert has the worst success rate, but is certain of all correct decisions and (justifiably) unsure of all incorrect decisions.

This leaves a certain degree of subjectivity in our choice of quantification. Rather than attempt to enforce a particular standard, we instead simply show how the validation results can be quantified and used for vetting a network's acceptance into a 'committee' (discussed subsequently). We leave the choice of what standard to enforce as a parameter in our general mechanism, to be chosen by the individual researcher.

If the network's outputs are rounded to 0 or 1, to enforce a decision, then the errors (ignoring the sign) are either 0 (correct) or 1 (incorrect).

Therefore, the errors for the n_v validation works come from a binomial distribution; we can estimate the mean and standard deviation as

$$\bar{e} = p = \frac{1}{n_v}\sum_{k=1}^{n_v} e_k, \qquad \sigma = \frac{p(1-p)}{\sqrt{n_v}}$$

and, consequently, calculate a confidence interval for the mean and/or perform a hypothesis test to determine whether the distribution differs significantly from some prescribed standard. For example, Merriam and Matthews (1993; 1994) compare the success of their validation predictions to random guessing ('coin tossing'). The sign of the errors (± 1) is also important: we want the network to be unbiased and, therefore, incorrectly predict Author 0 as often as incorrectly predicting Author 1. Since the sign of the error also comes from a binomial distribution, we can use the above procedure to test the likelihood of the distribution not having a true mean of 0. If there is significant evidence against the null hypothesis, then we should reject the network's predictions as being systematically biased towards one author. However, when dealing with limited validation data and enforcing a rigorous validation standard, there will be only a small number of misclassified validation works; it is therefore often the case that we do not have a large enough sample to achieve significant evidence against the null hypothesis. This is not a problem if a high validation standard is required and the validation data is reasonably equally distributed between both authors.

If we do not round the ouputs, then the errors (again ignoring the sign) are continuously distributed between 0 (completely correct) and 1 (completely incorrect). They therefore come from a β-distribution and the mean, μ, and standard deviation, σ, can be estimated by the sample mean and deviation, from which we can estimate the β-distribution parameters

$$\alpha = \frac{\mu}{\sigma^2}\left(\mu - \mu^2 - \sigma^2\right), \quad \beta = \frac{1-\mu}{\sigma^2}\left(\mu - \mu^2 - \sigma^2\right).$$

Using these parameters, we can then calculate confidence intervals and p-values for hypothesis tests. As in the rounded case, the signs of the errors again come from a binomial distribution, ideally with a mean of 0.

2.5 Committee and consensus

We know that human experts often come to contradictory conclusions, even though they have all demonstrated their knowledge and ability in various ways (written examinations, peer-reviewed research, etc.). Similarly, it is entirely possible to create many ANNs that produce different predictions, even when using identical architectures and training data, and even when they all achieve acceptably high validation success rates. The high dimensionality of the input data and the large number of degrees of freedom in the fitting function allow for many different functions that successfully fit both the training and validation data. Therefore, even the quantitative validation measures discussed above do not provide a failsafe measure of network 'correctness' (no more than an earned PhD guarantees human infallibility). Following a human analogy, then, we consider creating a committee of networks: having determined architecture, training works, etc., we train the network from random initial weights, calculate a measure of its validation success rate, and if this measure is sufficiently high, we 'admit the network to the committee' and record its predictions regarding the unattributed works.

After assembling a committee of a predetermined size, N_c, we can consider the distribution of the entire committee's predictions. As with the validation process, we can consider either the distribution of decisions (network outputs rounded to 0 or 1) or the actual values; again, the former gives a sample from a binomial distribution, the latter from a β-distribution. If this distribution indicates a consensus in the committee then we may make a claim about the authorship of the unknown works with greater confidence than when using only a single network. Furthermore, we may also make a 'non-prediction' in the case that no clear consensus is reached (such as a hypothesis test showing that the committee's decisions are not statistically different from random guessing—i.e. the committee is split roughly fifty-fifty, so the confidence interval for the proportion of networks predicting Author 0 contains the value 0.5).

Merriam and Matthews (1993; 1994) perform detailed analysis of the validation and predictions of their networks, and even generously offer to

provide their executable file to interested researchers. However, since their network is already fully trained, it will, of course, always give the same results as they presented. They quite correctly conclude that their networks' predictions should be considered seriously, but there is no way to know whether their network is representative of all possible networks. Indeed, their network would be admitted as a valid member of a committee, but by assembling a committee we can verify if that one particular network is aligned with the 'mainstream' view or is something of a 'renegade'.

2.6 Stylistic measures

Bearing in mind the old computing maxim 'garbage in, garbage out', we must carefully consider what to use as the inputs to the network. As previously noted, the list of possible statistics of a sample text is limited only by the statistician's imagination. (In the past, it might also be limited by the effort required in the compilation and calculation of the statistics; however, modern computing power allows for quick and automated calculation of stylistic statistics, particularly with the use of a scripting language such as *Perl* that permits complex text searching.) We therefore desire a way to reduce the list of input metrics to those with the greatest discriminatory power over the output. In previous studies—both using traditional stylometric methods and using ANNs—this has been done using linear statistical techniques such as multivariate regression and principal component analysis (PCA). ANNs, however, are an inherently nonlinear tool. In fact, the entire point of using ANNs, rather than a simple regression model, is to allow for decisions based on nonlinear interactions between the inputs.

For example, if we were to consider two metrics (say, average sentence length and word length), a traditional regression approach would evaluate the correlation of the output with each metric independently; but perhaps both authors under consideration use a wide range of word and sentence lengths, with their averages coming out very similar, in which case neither metric would show a significant correlation. However, it could be that one author uses short words more in short sentences and long words in long sentences, while the other author tends to mix (short words in long sentences and vice versa). In this case, these metrics are sufficient to explain the variation in the data, but only in a nonlinear manner, because it is the interaction of the metrics that is the fundamental discriminator between the authors. Thus, an ANN would be a perfect candidate for such a problem, but we should not use traditional statistical methods to determine which inputs to use.

PCA reduces the input dimension by constructing a new basis from linear combinations of the original variables. Although this therefore considers 'interactions' between the inputs, it does not avoid the problem discussed above, since the process is still linear. However, it is perfectly reasonable to use PCA to reduce the dimensionality after having chosen the input metrics; essentially this simply adds an initial linear level to the overall fitted function. We have done this in all the studies presented here; after selecting a subset of metrics to consider, linear combinations of those metrics are ranked according to how much variation of the data they explain, then we keep only the set that, collectively, explains 95% of the variation. By doing this we reduce the number of parameters in the fitted function and, therefore, we also reduce the computational complexity of training the network. The final inputs to the ANN are, therefore, not purely the chosen metrics, but linear combinations thereof. However, this PCA preprocessing is a purely linear manipulation and does not effect the ANN's ability to determine nonlinear correlations with the chosen metrics.

A number of new techniques are being developed in fields related to data mining and representation that could prove useful in situations such as this. Typically these involve trying to find a low-dimensional representation of data embedded in higher dimensions. We believe these should be considered extremely important avenues for further research; currently we know of no simple efficient algorithm for determining whether or not a particular input variable or dimension is useful for explaining the variation in the data in a nonlinear manner.

In the previous investigations of authorship using ANNs (Matthews and Merriam, 1993; Merriam and Matthews, 1994; Lowe and Matthews, 1995; Tweedie *et al.*, 1996), a handful of input metrics were used

that had been previously shown to have some discriminatory power. Naturally, this led to networks that worked well, demonstrating to some degree the utility of this approach. However, a reasonable objection is that 'the game was rigged': the networks used preselected metrics, but how can the correct metrics be determined (preferably in an automated way) for a different problem? Without having a simple solution to this, we proceed stochastically, selecting from a large list of 'probable' candidates. To assemble this list, we use: metrics that have been well investigated in terms of their general applicability (Holmes, 1994); variants of these; and a number of less well-established metrics, including some created for the sake of investigation. The full list is:

(1) average sentence length (words per sentence),
(2) possessive apostrophes per sentence,
(3) possessive apostrophes per sentence, averaged by sentence length,
(4) quotation marks per sentence,
(5) quotation marks per sentence, averaged by sentence length,
(6) dashes per sentence,
(7) dashes per sentence, averaged by sentence length,
(8) semicolons per sentence,
(9) semicolons per sentence, averaged by sentence length,
(10) commas per sentence,
(11) commas per sentence, averaged by sentence length,
(12) average word length,
(13) *no*/T10,
(14) *so*/T10,
(15) *with*/T10,
(16) *of X and*/*of*,
(17) *the X and*/*the* (where *the* is either *the* or *th'*),
(18) *no*/(*no* + *not*),
(19) *upon*/(*on* + *upon*),
(20) type-token ratio (number of different words/number of words),
(21) Yule's Characteristic $K = 100D(1 - 1/N)$, where D is Simpson's Index,
(22) entropy: $\sum_i - p_i \log(p_i)$ where $p_i =$ (number of occurrences of word i)/(total number of words),
(23) word-frequency distribution characteristic A,
(24) word-frequency distribution characteristic X,
(25) hapax legomena,
(26) hapax dislegomena,
(27) mixture measure (weighted average of number of times words on either side of a word from V_1 are used),
(28) average percentage of sentence position of T10 words,
(29) average spacing of the letter *e*,
(30) average spacing of the letter *m*,
(31) average spacing of the letter *o*, and
(32) average spacing of the letter *t*.

Metrics 2–11 measure punctuation usage; since usage of most of these punctuation marks was not standardized until relatively recently, these metrics would generally be considered unreliable for many applications. We include them because they may be useful in some situations and they therefore provide an interesting point for investigation. Each has two variants: the average number of specific punctuation marks per sentence, and this measure averaged by the sentence length (being equivalent to the average number of punctuation uses per word, averaged by sentences). Metric 1 (sentence length) is not a punctuation measure, *per se*, but requires some definition of what constitutes a sentence; typically, of course, this is punctuation-based. However, sentence termination is typically more uniformly defined than more modern punctuation marks such as dashes and semicolons. Nonetheless, whether a particular metric should be excluded as unreliable and the details of how to define and calculate each metric are stylometric issues that our algorithm cannot attempt to address automatically. We believe, however, that our stochastic approach could be used specifically to investigate such questions by including variants of a particular metric in the list of possibilities and seeing which variants are successfully used more than others. (More details of such an approach, but using only the above thirty-two metrics, are given in Section 4.)

Metrics 13–19 are all ratios of key common words; in the definitions given above, 'X' is any word and 'T10' is any of the ten function words of Wells and Taylor (1987): *but*, *by*, *for*, *no*, *not*, *so*, *that*, *the*,

to, *with*. The definitions of metrics 20–24 are given by Holmes (1994). Hapax legomena and dislegomena are the number of words used exactly once or twice, respectively, in the given sample.

Metrics 27 and 28 were invented for the sake of investigation. All the words in a sample can be sorted into disjoint sets V_k, consisting of all words used exactly k times; the 'mixture measure' (metric 27) is a weighted average of the six values of k corresponding to the three words on either side of any uniquely used word (i.e. a word from the set V_1). Hence, if an author typically uses unusual words (from V_1) mixed in with common words (large k), the value of this metric will be large; if the author uses blocks of uncommon words (low values of k), this value will be small. Metric 28 is deliberately somewhat arbitrary, but measures the location within a sentence of common function words (T10): if these words tend to be used at the beginning of sentences, this value will be low, and it will be high if the words are used more at the end of sentences. We readily acknowledge that, a priori, we do not expect this particular metric to be of any great use.

The last four metrics are all variations on a theme: the average letter spacing of the letter *e*—i.e. the average number of (nonpunctuation) characters between consecutive occurrences of the letter *e*—was, completely anecdotally, claimed to have possible discrimination power, based solely on a partially remembered undergraduate statistics project on regression! The letters *m*, *o*, and *t* were chosen semi-randomly: they happen to be the initials of one of the authors and also have high usage rates (like the letter *e*). This set of thirty-two metrics is only a selection of the many possible metrics that have been used in stylometric studies (not to mention the many more that could be considered—at very least, another twenty-two letter spacings are available beyond *e*, *m*, *o*, and *t*). However, in order to make the problem numerically feasible at this time, it is necessary to enforce some restriction of this form. One of the key avenues for ongoing research is that of increasing the computational efficiency of winnowing the input set, thus allowing for a wider range of metrics to be considered initially.

3 Algorithm

Fig. 2 shows the algorithm we use to assemble a committee of expert networks.

Steps 14 and 15 prevent the committee from constituting 'experts' of the same persuasion (same input metrics, trained on the same works); thus the committee is as diverse as possible. Furthermore, if we allow only one use of each set of metrics and training data, we can investigate which metrics and which works contribute most to successful training of the networks. In this way, we can rank the input metrics; we can also see if any canonical works are problematic ('out of character')—ideally we would see a uniform distribution of the training works.

Steps 11a and b use the general notion that too many parameters typically lead to overfitting, whereas too few lead to underfitting. In the former situation, we expect to see the network converge but fail validation; in the latter case, we would see the network fail to train. Step 4 prevents 'flogging a dead horse' in the case that we happen to have chosen a bad combination of metrics and/or training data.

This algorithm requires a number of parameters to be set; steps 3, 4, 10, 11, 14, and 15 all require a specific parameter, and steps 7 and 9 require some definition of success. In the experiments presented here: we reset the randomly selected metrics and training data (steps 14 and 15) every time they are used; start with n_h (step 3) equal to the number of input variables; reset the initial random weights (and retrain) until the number of failed trainings (step 10b) or validations (step 10a) is three times the number of successes; increase n_h (step 11b) by a factor of 1.5 and decrease it (step 11a) by a factor of 0.8 (but at least 1 in either case); and limit n_h (step 4) to be between 1 and 50. For validation, we require the network to pass two tests: achieve at least 85% accuracy on the decision for each sample, and correctly predict every full work (consisting of several samples) after the average of all samples is taken across the entire work. For the former test, we in fact require that the decisions come from a distribution where the mean success

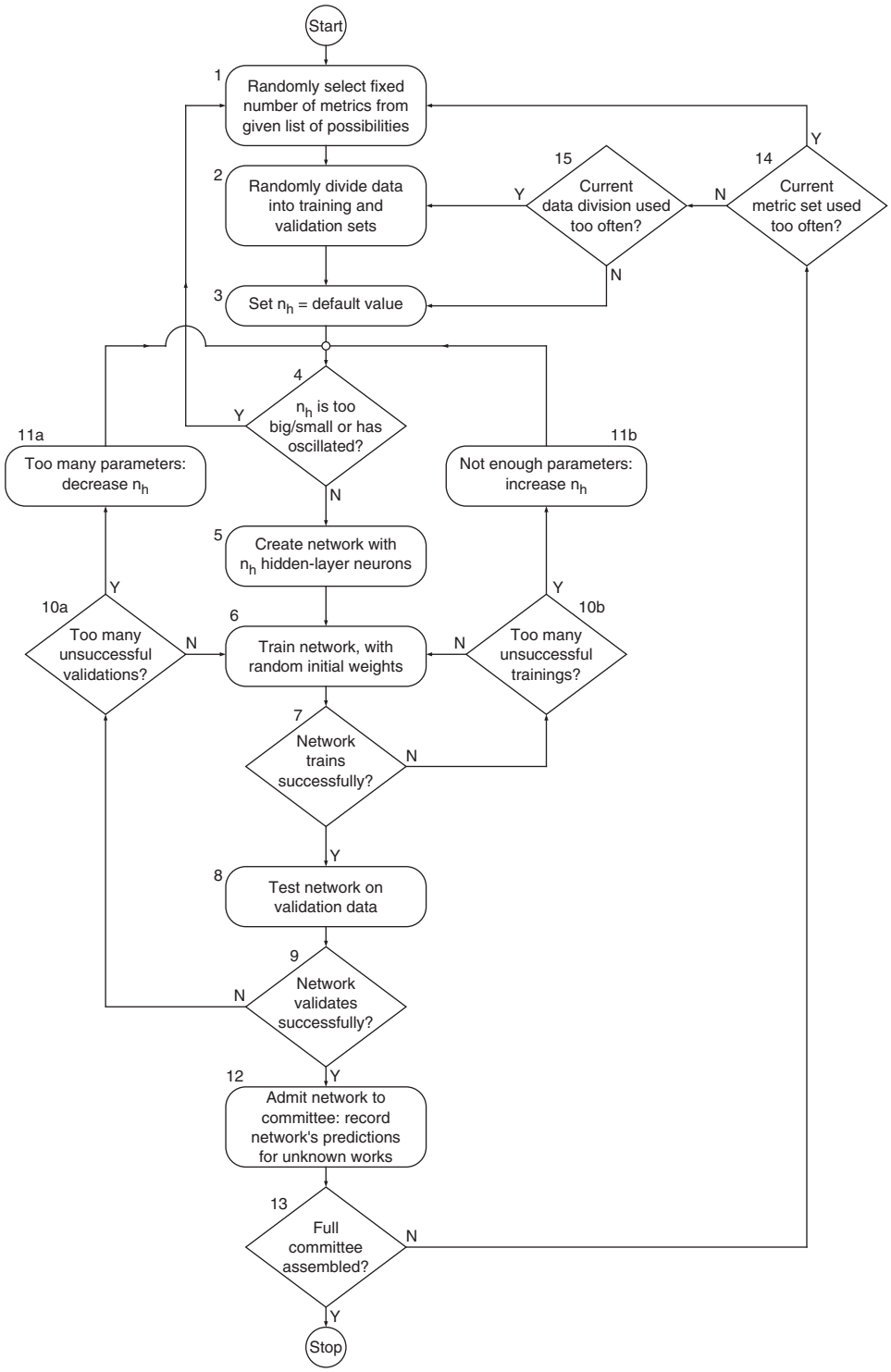

Fig. 2 Flowchart for our algorithm to stochastically assemble committees of 'expert' neural networks

rate is greater than 85%, to 99% confidence (i.e. the lower bound on the 99% confidence interval for the mean is greater than 0.85). For the latter test, we average the actual prediction values across a given work, so that the network's 'confidence' in its predictions for individual samples is taken into account (thus eliminating, for example, a network that 'equivocates'—predicts near 0.5—for most samples and is certain, but completely wrong, for the remaining few).

Although there are many parameters in use here and they do affect the computation time, they do not affect the robustness of the algorithm. The effect of certain parameters on computation time is obvious: more rigorous training/validation requirements will lead to longer computations, as networks that would have been adequate for less stringent requirements are rejected, and reselecting the metrics and training/validation division every time also extends run time.

Our code was written in MatLab and makes use of its Neural Network Toolbox. Although not the fastest computational language, using MATLAB enabled us to perform a number of experiments easily, thus providing us with an environment for investigation into the concept of a general network, without needing to code the neural network elements ourselves. The experiments we ran typically required anything from a few hours to a week or more of total computer time.

4 Results

In order to demonstrate the generality of our algorithm, we apply it to two separate authorship pairings, of different time periods and styles. First, we consider arguably the most famous authorship pairing of all: William Shakespeare (1564–1616) and Christopher Marlowe (1564–1593). We then look at The *Federalist Papers*, eighty-five essays written by John Jay (1745–1829), Alexander Hamilton (1755–1804), and James Madison (1751–1836). These two cases were chosen because they have been studied previously both with traditional stylometric methods and also ANNs. Although the validation step in our algorithm requires that the networks in our committees have already demonstrated their ability to generalize, for the purposes of testing our algorithm we will consider 'unknown' works that are, in fact, undisputed. Having demonstrated our algorithm's success with a manufactured test, we will present results for genuinely ambiguous cases in future work.

4.1 Shakespeare and Marlowe

Choosing randomly, we consider *As You Like It*, *Cymbeline*, *King Lear*, *The Merry Wives of Windsor*, *Tamburlaine, Part II*, *The Tempest*, *Titus Andronicus*, and *Troilus And Cressida* as our 'unattributed' works, with all other standard canonical works as our training and validation data. (The random selection was, however, performed until at least one work of Marlowe's was included in the set.) The inclusion of *Titus* is fortuitous, since Merriam and Matthews (1994) found an anomalous stylometric signature for this work; they state that *Titus* is 'closely related to an early quarto version, and its purely Shakesperian nature remains doubted'.

The uneven distribution of authorship among the test works provides a significant challenge for our method. By varying the author distribution of the training and validation works, we were able to verify that this distribution affects the distribution of the networks' predictions: a network that is trained more heavily on Shakespeare than Marlowe will be more likely to 'play the percentages' and vote more towards Shakespeare. By enforcing equal training/validation data, we—in theory—remove this bias. A committee that then correctly predicts an unequal test set, such as we have, must indeed be recognizing genuine stylometric patterns, rather than a simple volume bias.

We used *Perl* to parse electronic versions of the texts (using the Collins edition Shakespeare) and calculate the thirty-two metrics listed above for samples of 1000 words. In the results shown here, punctuation metrics (2–11) were not included, as these are unreliable for Elizabethan works. We do, however, allow for sentence length (metric 1); although this is punctuation based, we consider it to be the least ambiguous of all punctuation metrics. Interestingly, when we included all metrics, the algorithm

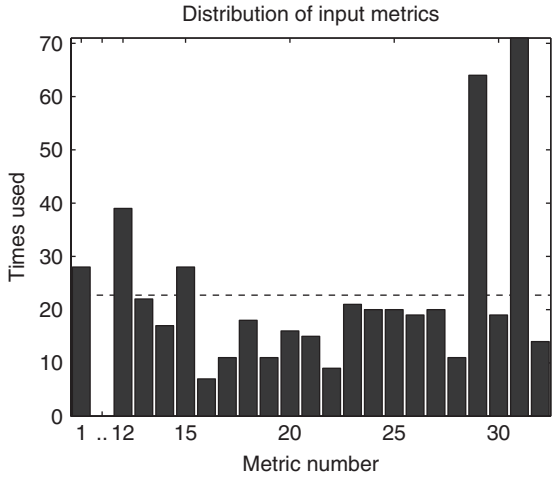

Fig. 3 Frequency of input metric usage for a committee of 100 5-input networks; the dashed line represents a uniform distribution, adjusted for the removed metrics

worked extremely well and quickly, and generally favored the punctuation metrics (five of the top six in usage coming from metrics 2–11); sentence length, however, was not particularly highly ranked. Although this seems counter-intuitive to our decision to exclude punctuation metrics, we believe that this in fact demonstrates their unreliability: the networks' decisions actually reflect the latter-day editors who have compiled these works, rather than the original authors themselves.

Fig. 3 shows the distribution of the usage frequency of input metrics for networks successfully auditioned into a committee of 100 members. Each network used $m = 5$ inputs at a time. The distribution is clearly nonuniform. There are five metrics that are used more frequently than average (and only three significantly more): sentence length (metric 1), word length (metric 12), *with*/T10 (metric 15), and *e*- and *o*-spacing (metrics 29 and 31). This list is somewhat surprising, given that only metric 15—which, moreover, ranks last of the five—is generally considered a 'good' metric (Holmes, 1994). (Sixth is *no*/T10, used slightly less frequently than average.) However, as is demonstrated by the case of Oliver Goldsmith (Mannion and Dixon, 2004), sometimes metrics with limited power in general have great power in a specific instance. This, then, is one of the benefits of our approach: the most useful inputs for a specific problem are automatically determined.

Applying these committees to the specific questions of works of uncertain origin, we obtain Fig. 4. The committee's predictions for each work form a β-distribution between 0 (Shakespeare) and 1 (Marlowe). Clearly the committee is correct in each case, with all works being attributed—in consensus—to Shakespeare, with the exception of the Marlovian *Tamburlaine*. Furthermore, the most contentious decision is that of *Titus Andronicus*; as discussed above, this is in perfect agreement with previous studies. The exact values of the means and proportions of the committee's votes are given in Table 1.

In our algorithm, the input metrics are randomly selected in groups of m (out of the permitted twenty-two metrics). Although Fig. 3 shows that certain metrics are used in almost all successful networks, in any given network the metrics are likely to be a mix of frequently used and infrequently used. Assuming that higher frequency of use is due to greater discriminatory power, then it is reasonable to consider restricting ourselves to using only the m most frequently used metrics. We therefore restrict the allowable metrics to the set $\{1, 12, 15, 29, 31\}$. Although we now essentially bypass the random selection of inputs, we retain all the other features of our algorithm. Since we do not need to experiment with the input metrics, the algorithm runs faster, allowing us to assemble larger committees in reasonable times. The predictions from a committee of 500 networks are shown in Fig. 5 and the values are given in Table 2. As might be expected, using only a selected set of metrics leads to more homogeneous predictions from the committee: the means and proportions generally move slightly towards the correct value, and the variances are slightly decreased.

4.2 The Federalists

The eighty-five Federalist essays were written by John Jay (2–5, 64), Alexander Hamilton (1, 6–9, 11–13, 15–17, 21–36, 59–61, 65–85), and James Madison (10, 14, 37–48), with three (18–20) being written jointly by Hamilton and Madison, defending the fledgling Constitution. The remaining twelve

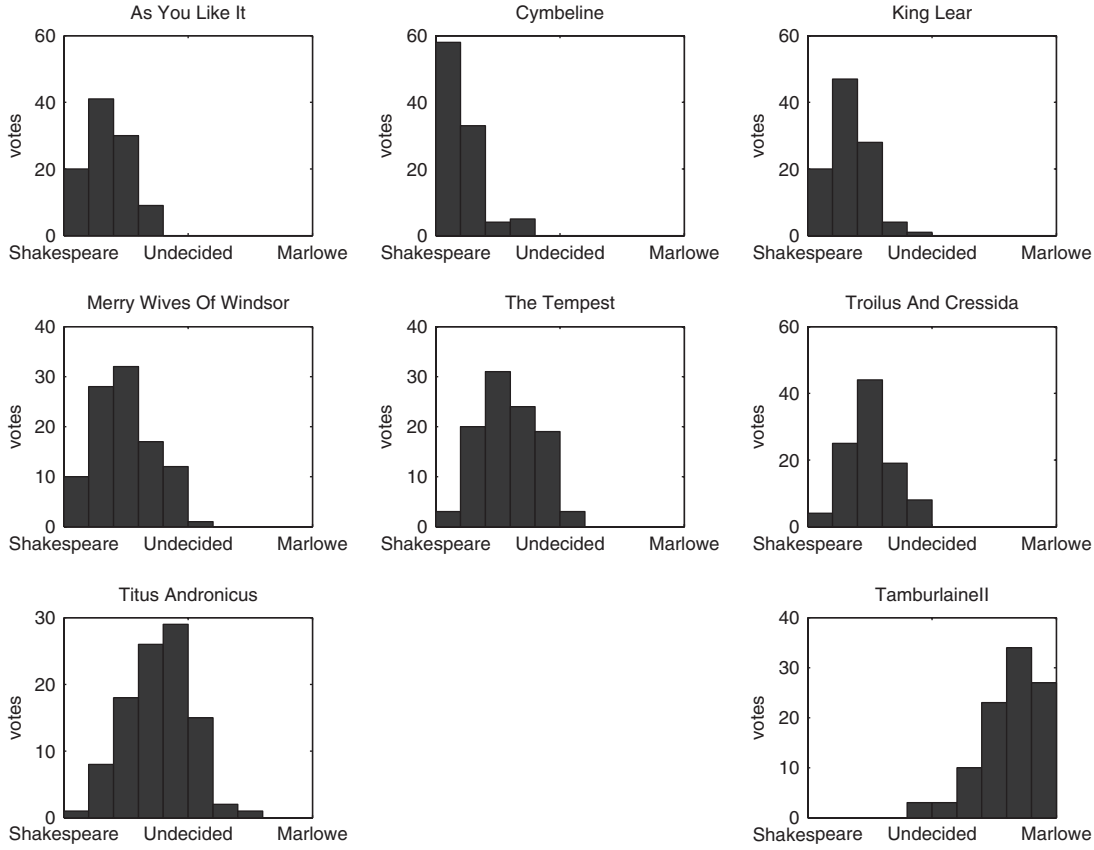

Fig. 4 Histogram of the predictions for 'questionable' works from the committee of 100 5-input networks used in Fig. 3

Table 1 Mean, μ, proportion, P, and 99% confidence interval (CI) for the mean of predictions of a committee of 100 5-input networks

Work	μ	P	CI
As You Like It	0.176	0	[0.027, 0.442]
Cymbeline	0.095	0	[0.001, 0.417]
King Lear	0.168	0	[0.023, 0.435]
The Merry Wives Of Windsor	0.247	0.01	[0.034, 0.598]
Tamburlaine	0.815	0.97	[0.409, 0.993]
The Tempest	0.294	0.03	[0.066, 0.617]
Titus Andronicus	0.382	0.18	[0.102, 0.725]
Troilus And Cressida	0.250	0	[0.066, 0.517]

A value of 0 corresponds to Shakespeare, and 1 to Marlowe.

essays (49–58, 62, 63) were claimed (*a posteriori*) by Hamilton even though they were generally attributed to Madison. As in the Shakespeare–Marlowe case, this dispute has been studied previously, with both traditional stylometric methods (Mosteller and Wallace, 1984; Merriam, 1989; Martindale and McKenzie, 1995) and ANNs (Tweedie *et al.*, 1996). Various studies have rejected Hamilton's claims, with paper 55 providing the only possible exception, and even that now appears unlikely. However, we again concentrate on undisputed works, rather than the disputed ones, in order to test our method; as with the Elizabethan playwrights, analysis of the genuinely disputed Federalist papers will be discussed in future work. Our 'disputed' essays were randomly chosen (again requiring at least one of each author) to be essays 6, 11, 22, 36, 37, 44, 69, 71, 75, 77, 79, and 82 (all written by Hamilton except 37 and 44).

Although only metric 19 (*upon*/(*on* + *upon*)) of our list has previously been used to advantage for

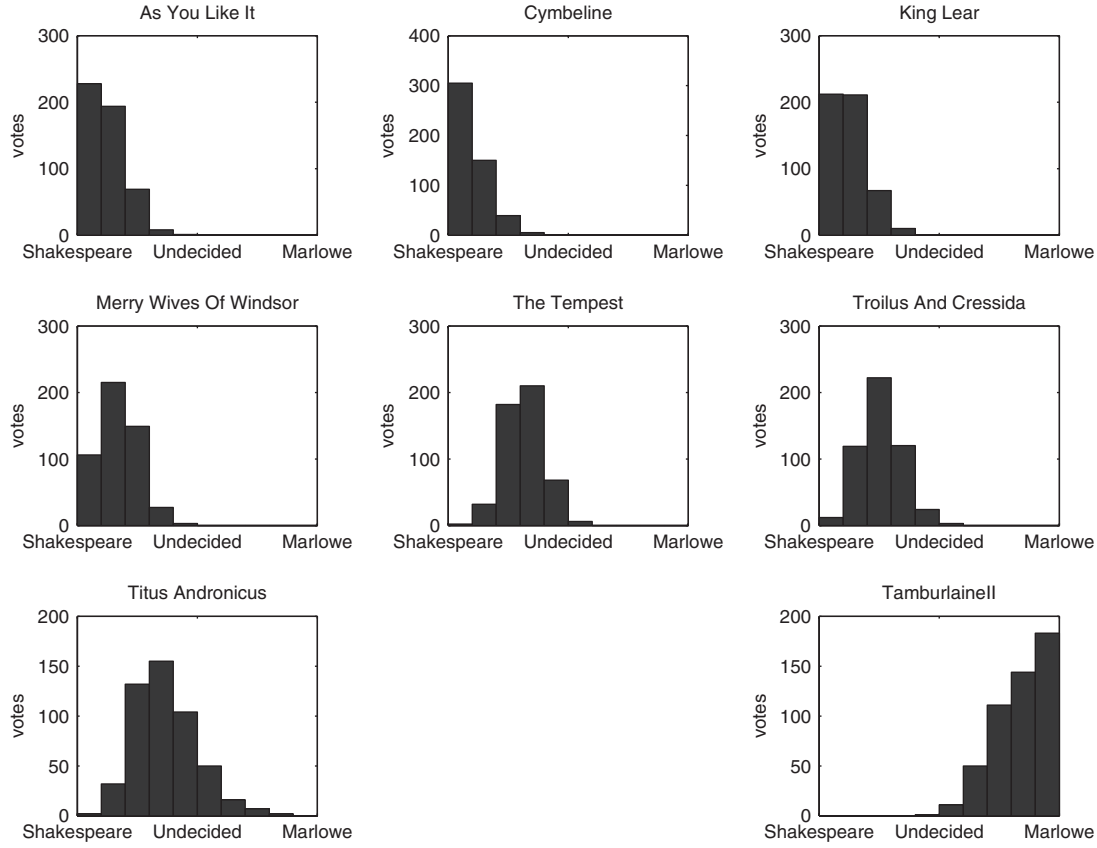

Fig. 5 Histogram of the predictions for 'questionable' works from a committee of 500 5-input networks, with the input metrics fixed

Table 2 Mean, μ, proportion, P, and 99% confidence interval (CI) for the mean of predictions of a committee of 500 5-input networks, with fixed input metrics

Work	μ	P	CI
As You Like It	0.124	0	[0.008, 0.392]
Cymbeline	0.085	0	[0.000, 0.399]
King Lear	0.122	0	[0.008, 0.382]
The Merry Wives Of Windsor	0.169	0	[0.021, 0.449]
Tamburlaine	0.844	0.998	[0.445, 0.997]
The Tempest	0.315	0.012	[0.131, 0.543]
Titus Andronicus	0.367	0.15	[0.092, 0.714]
Troilus And Cressida	0.256	0.006	[0.078, 0.505]

A value of 0 corresponds to Shakespeare, and 1 to Marlowe.

this particular problem (Mosteller and Wallace, 1984; Merriam, 1989), we nevertheless apply our algorithm without alteration as a test of its generality. As before, we track which inputs the committee 'chose' from the list (including punctuation metrics, in this case). The only difference in algorithm between this example and the previous example with Shakespeare and Marlowe is that the works here are much shorter—each on the order of 1000 words, which was the size of the samples used for the Elizabethan plays. We therefore do not divide the individual essays into parts, but simply compute the statistics for each essay whole.

Taking $m = 5$ inputs at a time, Fig. 6 shows that metric 19 is used in every successfully auditioned network, in complete agreement with the findings of Merriam (1989). The only other metrics used significantly above average are metrics 22 (entropy) and 13 (*no*/T10) and there are only five other metrics used even slightly more often than average. The predictions from this committee of 100 networks,

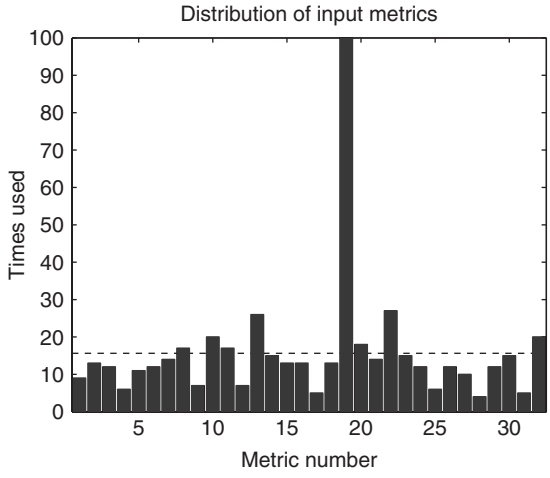

Fig. 6 Frequency of input metric usage for a committee of 100 5-input networks; the dashed line represents a uniform distribution

shown in Fig. 7 and Table 3, are correct (in consensus) in every instance. As before, we use the highest frequency metrics to create networks with fixed input metrics, allowing us to create larger committees. The predictions from a 500-network committee with five inputs—metrics 10 (commas per sentence), 13, 19, 22, and 32 (*t*-spacing)—are shown in Fig. 8 and Table 3. As before, there is little difference between the predictions of the two committees, with the committee of fixed input metrics being slightly more unified in its consensus.

An interesting facet of the Federalist predictions is that the distributions are far more polarized—more like a binomial distribution than a β-distribution. This leads to difficulties in calculating a meaningful confidence interval. Clearly, however, there is no question that the correct consensus has been reached in each of the Federalist test cases.

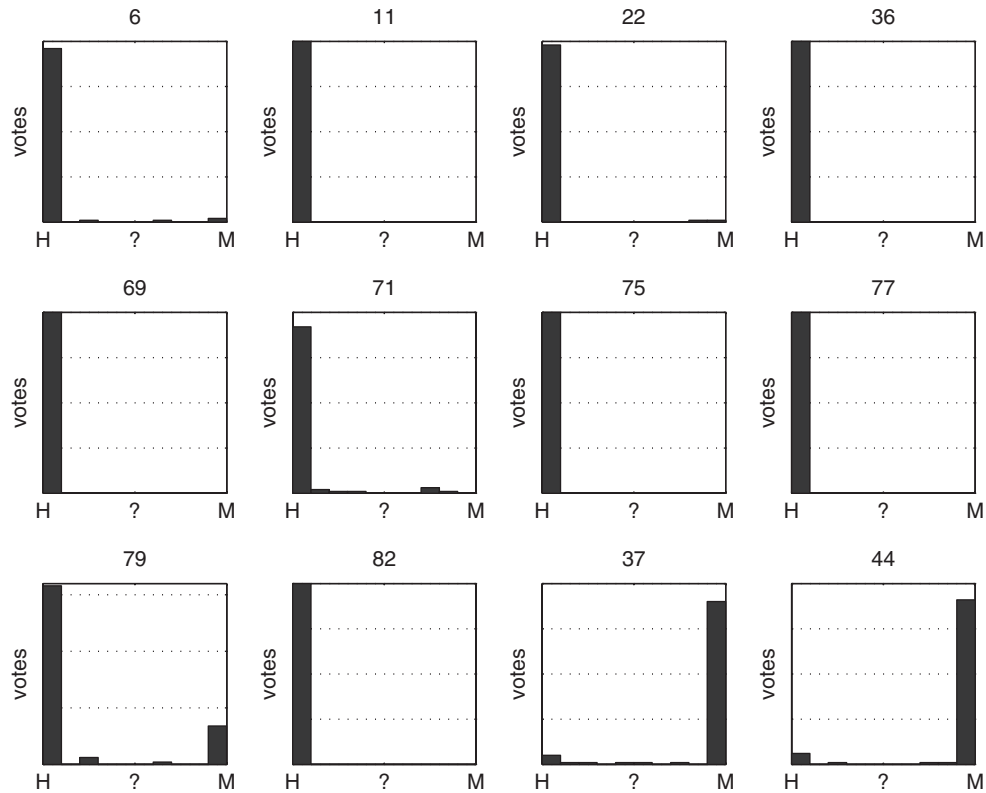

Fig. 7 Histograms of predictions for 'disputed' works from a committee of 100 5-input networks. 'H' is Hamilton, 'M' is Madison, and '?' is undecided. For clarity the vertical scale is omitted; each scale ranges from 0 to 100

Table 3 Mean, μ, and proportion, P of predictions of a committees of 5-input networks; the first two values are for a committee of 100 networks with various input metrics; the second two are for a committee of 500 networks with fixed input metrics

Essay	μ	P	μ	P
6	0.032	0.03	0.003	0.002
11	0.002	0	0.001	0
22	0.020	0.02	0.005	0.002
36	0.001	0	0.004	0.002
37	0.919	0.92	0.921	0.928
44	0.928	0.93	0.983	0.988
69	0.001	0	0.002	0
71	0.048	0.04	0.011	0.008
75	0.002	0	0.003	0.002
77	0.001	0	0.001	0
79	0.186	0.18	0.039	0.03
82	0.001	0	0.002	0.002

A value of 0 corresponds to Hamilton, and 1 to Madison.

Furthermore, even in the Shakespeare–Marlowe case, many of the 99% confidence intervals include 0.5, meaning that technically there is not sufficient evidence against the null hypothesis of a 'non-decision', at least at the 99% confidence level. Again, however, it is clear in these cases that the consensus of the committee is strongly in favor of a decision (and, as demonstrated above, the correct one). Thus one final possible area for further investigation is how to work with such variable distributions and how to define a reasonable criterion for what constitutes a definitive consensus prediction.

Finally, it should be noted that the predictive power of metric 19 was discovered by Mosteller and Wallace (1984) using traditional statistical methods. Its inclusion in our list of thirty-two possible metrics was due to its use in previous studies, but this particular metric would not normally

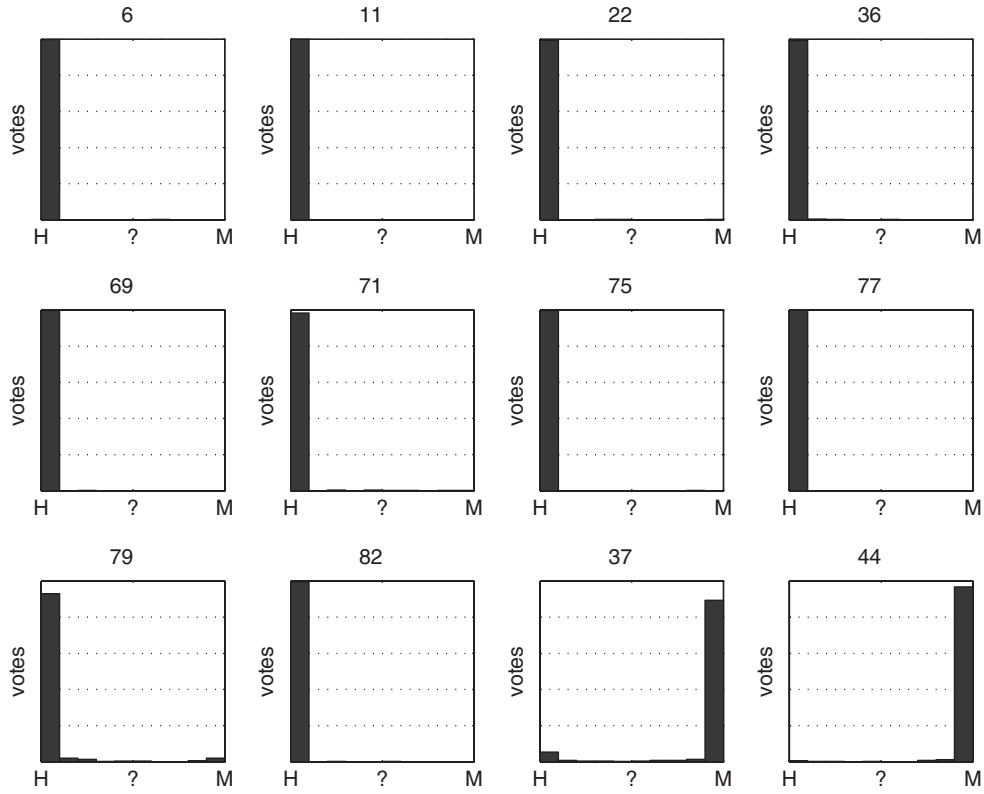

Fig. 8 Histograms of predictions for 'disputed' works from a committee of 500 5-input networks with fixed input metrics. 'H' is Hamilton, 'M' is Madison, and '?' is undecided. For clarity the vertical scale is omitted; each scale ranges from 0 to 500

appear in a 'general' list. In this case, classical stylometric methods were sufficient to uncover a correlation; as noted above, we know of no currently available method by which general nonlinear correlations can be determined efficiently. As demonstrated here, randomly selecting the input metrics can achieve this to some degree by revealing which metrics are most often used by successful networks. Our algorithm could therefore be used for this purpose, using a specific list of possible metrics to choose from. Determining an automated way to generate this list of 'likely' metrics is an intriguing future research direction.

Both test problems shown here (Shakespeare–Marlowe and the Federalists) were for binary classification problems. Initial experiments indicate that this approach can also work for multi-author problems, and possibly even a binary classification of 'Author X' and 'not Author X' (i.e. a sample of various authors). However, multiple author classification inevitably increases computational complexity further. We therefore leave this avenue unexplored until issues of efficiency are more satisfactorily resolved.

5 Conclusion

Considering the problem of developing a truly automated method for determining authorship of unattributed or disputed works, we have presented an algorithm that uses random permutations to select input metrics from a list of possibilities, and also to select training and validation works from the core canon, then trains and auditions networks in order to assemble a committee that can make predictions on the unattributed works.

The advantage of this approach is that it does not require prior information or calculation regarding the discriminatory power of the input metrics. Rather, it allows the nonlinear nature of ANNs itself to reveal correlations in the training data. Currently, the algorithm is computationally intensive; we hope that further research will provide continuing improvements in efficiency. Indeed, we believe that the current work highlights an important immediate avenue of research in this area: input selection; that is, how to determine, from a limited data set, which set of variables explains the greatest variation in the data in a general, nonlinear manner.

We applied our algorithm to two famous attribution problems: Shakespeare and Marlowe, and The Federalist Papers. In the former case, although the input metrics that were most useful were unexpected, the predictions from our network committee were correct in every case, even highlighting the problematic nature of *Titus Andronicus*. Once the networks were trained to distinguish between Shakespeare and Marlowe, they were able to answer several questions of disputed authorship simultaneously. In the case of the Federalists, our algorithm demonstrated a strong correlation between one metric (*upon*/(*on* + *upon*)) and the resulting author, in agreement with previous studies (Mosteller and Wallace, 1984; Merriam, 1989). Again, once trained, our networks considered all questionable works simultaneously, returning perfect results. Results regarding genuinely disputed works for both problems will be presented in future work.

By successfully applying the same algorithm, with virtually no human guidance, to two different problems of the same general pattern, we have provided an initial step towards an important goal in literary computation: a general mechanism for stylometric authorship attribution.

References

Damerau, F. J. (1975). The Use of Function Word Frequencies as Indicators of Style. *Computers and the Humanities*, **9:** 271–80.

de Vel, O., Anderson, A., Corney, M., and Mohay, G. (2001). Mining E-mail Content for Author Identification Forensics. *SIGMOD Record*, **30:** 55–64.

Elliott, W. E. Y. and Valenza, R. J. (1996). And then there were none: Winnowing the Shakespeare Claimants. *Computers and the Humanities*, **30:** 191–45.

Foster, D. W. (1996). Response to Elliott and Valenza, 'And then there were none'. *Computers and the Humanities*, **30:** 247–55.

Hagan, M. T., Demuth, H. B., and Beale, M. (1996). *Neural Network Design*. Boston: PWS Pub. Co.

Holmes, D. I. (1994). Authorship Attribution. *Computers and the Humanities*, **28:** 87–106.

Lowe, D. and Matthews, R. (1995). Shakespeare vs. Fletcher: A Stylometric Analysis by Radial Basis Functions. *Computers and the Humanities*, **29:** 449–61.

Mannion, D. and Dixon, P. (2004). Sentence-length and Authorship Attribution: the Case of Oliver Goldsmith. *Literary and Linguistic Computing*, **19:** 497–508.

Martindale, C. and McKenzie, D. (1995). On the Utility of Content Analysis in Author Attribution: *The Federalist*. *Computers and the Humanities*, **29:** 259–70.

Matthews, R. A. J. and Merriam, T. V. N. (1993). Neural Computation in Stylometry I: An Application to the Works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, **8:** 203–9.

Merriam, T. (1989). An Experiment with the Federalist Papers. *Computers and the Humanities*, **23:** 251–54.

Merriam, T. (2003). Intertextual Distances, Three Authors. *Literary and Linguistic Computing*, **18:** 379–88.

Merriam, T. V. N. and Matthews, R. A. J. (1994). Neural Computation in Stylometry II: An Application to the Works of Shakespeare and Marlowe. *Literary and Linguistic Computing*, **9:** 1–6.

Milic, L. (1991). Progress in Stylistics: Theory, Statistics, Computers. *Computers and the Humanities*, **25:** 393–400.

Mosteller, F. and Wallace, D. L. (1984). *Applied Bayesian and Classical Inference: The Case of the Federalist papers*, New York: Springer-Verlag.

Potter, R. G. (1991). Statistical Analysis of Literature: A Retrospective on *Computers and the Humanities*, 1966–1990. *Computers and the Humanities*, **25:** 401–29.

Singh, S. and Tweedie, F. (1995). Neural networks and disputed authorship: New challenges, In *Artificial Neural Networks*. IEE, Conference Publication No. 409, pp. 24–28.

Tweedie, F. J., Singh, S., and Holmes, D. I. (1996). Neural Network Applications in Stylometry: The *Federalist Papers*. *Computers and the Humanities*, **30:** 1–10.

Valenza, R. J. (1991). Are the Thisted-Efron Authorship Tests Valid? *Computers and the Humanities*, **25:** 27–46.

Wells, S. and Taylor, G. (1987). The canon and chronology of Shakespeare's Plays. In *William Shakespeare: A Textual Companion*. Oxford: Clarendon Press, pp. 69–144.

Zheng, R., Qin, Y., Huang, Z., and Chen, H. (2003). Authorship Analysis in Cybercrime Investigation. *Lecture Notes in Computer Science*, **2665:** 59–73.