Ant Colony Optimisation for Stylometry: The Federalist Papers.

Michael P. Oakes
School of Computing and Technology
University of Sunderland
DGIC, St. Peter's Campus, Sunderland SR6 0DD, United Kingdom
e-mail: Michael.Oakes@sunderland.ac.uk

Abstract*: This paper describes the use of Ant Colony Optimisation for the classification of works of disputed authorship, in this case the Federalist Papers.Classification accuracy was 79.1%, which compares reasonably well with previous work on the same data set using neural networks and genetic algorithms. Although statistical approaches have performed much better than this, the advantage of a rule-based approach is the ability to produce readily intelligible criteria for the classification decisions made*.

Keywords*: swarm intelligence, .ant colony optimization, stylometry, Federalist Papers.*

1. Introduction: The Federalist Papers

Computational stylometry is the computer analysis of literary style, and includes the topic of automatic author attribution in cases of disputed authorship. The 85 Federalist papers, published under the pseudonym *Publius* in newspapers in 1787 and 1788 to persuade the people of New York State to ratify the new American Constitution, form an ideal test bed for stylometric investigations. 43 of the papers were undisputedly written by Alexander Hamilton, 14 by James Madison, 5 by John Jay and 3 jointly by Hamilton and Madison. The historical controversy arises because on the night before he was killed in a duel, Hamilton claimed in a list given to his friend Egbert Benson that he had written the 12 others. Madison did not dispute this at the time, but after having retired from the presidency, claimed that he himself was the true author of the papers on the Benson list. A number of authors have used computer stylometric techniques to try and resolve this case of disputed authorship. First Mosteller and Wallace [1] identified a list of 30 "marker" words, found to be more typical of either Hamilton or Madison's undisputed writings. They combined the evidence of how often each of these words occurred in each disputed paper using Bayes' rule, and found that the odds on each paper having been written by Madison were at least 240 to 1. The Federalist Papers have been used several times since as a testing ground for authorship attribution techniques [2]. They form a good test bed, because of the large number of undisputed writings by each author, and the fact that the disputed texts were claimed by only two authors. The writing styles of Hamilton and Madison are very similar, and thus techniques developed to distinguish between them should work well for other author pairs. Subsequent studies have included the following techniques:

1. Use of the log-likelihood ratio [5] and other univariate statistical measures [6].
2. Neural networks with back propagation. Tweedie, Holmes and Singh [9] used ten input nodes corresponding to 10 function words: *an, any, can, do, every, from, his, may, on, there* and *upon*. To avoid making any prior decision about which words might prove good discriminators, Kjell [3] used [26 * 26] input nodes, corresponding to each pair of adjacent characters which might appear in an English text. Both sets of authors used a hidden layer, and two output nodes: one corresponding to Madison, and one to Hamilton.
3. Holmes and Forsyth [2] used 5 measures of vocabulary richness, where a rich text would contain many different words per unit text length. They clustered the papers according to their richness scores by Principal Component Analysis, and found that all the disputed papers fell into the same cluster as the papers undisputedly written by Madison.

4. Holmes and Forsyth also used a genetic algorithm to learn decision rules, such as ((KIND < 0.00002) & (TO < 35.205)) → Madison, meaning that a disputed document was written by Madison if the word kind appears less than 0.00002 times per 1000 words, and to appears more than 35.205 times per 1000 words.

In this paper we will look another method of learning decision rules to distinguish between Madison and Hamilton, namely Ant Colony Optimisation.

2. Ant Colony Optimisation

Ant Colony Optimisation (ACO) is a form of swarm intelligence. When gathering food, ants lay down trails of a chemical called pheromone as a means of communicating with other ants. The likelihood that an individual ant will follow a given trail depends of the amount of pheromone already laid down upon that trail. All ants are assumed to move at the same speed and lay down pheromone at the same rate. Pheromone accumulates faster in the best paths, because the ants using the best paths will be able to move backwards and forwards from the food supply to the nest on more occasions per unit time than an ant using a less convenient route. Eventually all the ants converge to the shortest path, which will ideally be close to optimal. This is an example of positive feedback in nature. In Parpinelli et al.'s [8] Ant_Miner computer simulation, each path followed by an ant is associated with a candidate solution to a categorisation problem.

Rules are produced in the form "IF <condition> THEN <class>, where the condition is an AND-ed sequence of terms. Each term consists of the triple <attribute operator value>, such as <GENDER = FEMALE>. The only operator they allowed in the original Ant_Miner [8] was =, since all attributes were considered to be membership of a particular category. In the simulation described in this paper, the operators < and > were used, as the terms were word frequencies in a range of values, such as KIND < 469 per million words. The boundaries of the frequency ranges were the mean usages in the Federalist papers as a whole, and the means plus or minus 0.2, 0,4 … 3.0 standard deviations. Each attribute was one of Mosteller and Wallace's 30 marker words. Such rules provide knowledge which is intuitively comprehensive to the user, which is one of the goals of data mining. The rules are discovered from training data, then applied to test data. In this paper the training data is the Federalist papers of known authorship (excluding those by Jay and three jointly written papers), and the test data are the 12 papers of disputed authorship.

Ant_Miner simulates the pheromone updating as follows. First all cells in the pheromone table are initialized equally to the following value:

$$t_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i}$$

where a is the total number of attributes, bi is the number of values in the domain of attribute i. In the simulation described in this paper, each attribute can take one of 30 range values. The pheromone updating rule is as follows, where Q is a fitness score for each candidate rule.

$$t_{ij}(t+1) = t_{ij}(t) + t_{ij}(t).Q$$

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg}\right) x \left(\frac{TrueNeg}{FalsePos + TrueNeg}\right)$$

As well as being needed for the pheromone updating rule, Q is also used for identifying the best candidate rule found by each ant. It is also used for rule pruning, to determine whether a rule will work just as well or better if one component of a rule is taken out.

In real life, ants have very little vision, but in Ant_Miner an information theoretic vision heuristic is used. A simpler vision heuristic is used in Ant_Miner2 [4], which is claimed to work just as well. The Ant_Miner2 vision heuristic, used in this paper, is achieved by a problem dependent function that estimates the quality of terms to be added to the partial solution.

$$h_{ij} = \frac{majority\_classT_{ij}}{\left|T_{ij}\right|}$$

|Tij| is the set of training documents which match partial rule ij (such as word i occurs with a frequency in the range j). If there were 10 such documents, 8 by Madison and 2 by Hamilton, then the majority_classTij would be the maximum of 8 and 2 which is 8. In both Ant_Miner and Ant_Miner2, the vision heuristic function is short sighted, since it considers only one attribute at a time, and ignores attribute interactions. Real ants are also thought not to have memory, but in the simulations discussed here an element of memory is introduced, as convergence is achieved when a given number of consecutive ants follow the same path. The probability that a new term should be added to the rule currently being built is given below. This is the product of the amount of pheromone and the vision heuristic for each attribute-value pair, normalised by the sum of this product over all values of all attributes. I is the set of attributes that are not yet used by the rule being built.

$$P_{ij}(t) = \frac{t_{ij}(t).h_{ij}}{\sum_{i}^{a}\sum_{j}^{bi} t_{ij}(t).h_{ij}, \forall i \in I}$$

```
TrainingSet = {all training cases};
DiscoveredRuleList = [ ];  /* rule list is initially empty */
WHILE (TrainingSet > Max_uncovered_cases){
      t = 1; /* ant index */
      j = 1; /* convergence test index */
      Initialise all trails with the same amount of pheromone;
      REPEAT
            Ant[t] starts with an empty rule and incrementally constructs a
classification rule R[t] by adding one term at a time to the current rule;
            Prune rule R[t];
            Update the pheromone of all trails by increasing pheromone in
the trail followed by Ant[t] (proportional to the quality of R[t]) and
decreasing pheromone in the other trails (simulating pheromone
evaporation);
            IF (R[t] is equal to R[t-1] /* update convergence test */
                  THEN j = j + 1;
                  ELSE j = 1;
            END IF
            t = t + 1;
      UNTIL ( t >= No_of_ants) OR (j >= No_rules_converg)
      Choose the best rule R[best] among all rules R[t] constructed by all
the ants;
      Add R[best] to DiscoveredRuleList;
      TrainingSet = TrainingSet - {set of cases correctly covered by
R[best]};
END WHILE
```

Figure 1. A High-Level Description of Ant-Miner [8].

Rules are constructed by individual ants, by repeatedly combining terms until fewer than a certain value (the parameter minimum cases per rule) training set cases conform with this rule.

After rule construction comes rule pruning, a technique commonly used in data mining to remove irrelevant terms to give simplicity and avoid overfitting to the training data. A high level description of the entire Ant_Miner process is given in Figure 1.

3. ACO for Text Classification

A closely related topic to text classification by author is text classification by topic content, a task which was performed by Holden and Freitas [1] using another version of Ant_Miner. They made no prior assumptions which words in the web pages to be classified were to be used as potential discriminators. To reduce data sparseness, they used stemming, a technique whereby different grammatical forms of a root word can be considered equivalent, such as *borrow, borrowing and borrowed*. In the study described in this paper, a few of the 30 marker words were in fact related word pairs such as *considerable* and *considerably* considered as one. Holden and Freitas also conflated sets of words if they were closely related in the WordNet electronic thesaurus, so for example, sets of words related by links such as "broader than" (e.g. *cat* and *tabby*) could also be considered as one. The 30 marker words all occur relatively frequently in the Federalist papers, so data sparseness was not a problem in this study. Holden and Freitas compared Ant_Miner with the rule induction algorithms C4.5 and CN2. They found that Ant_Miner was comparable in accuracy, and formed simpler rules.

Table 1. Parameters of the Ant_Miner model.

Parameter	This study	Parpinelli et al. [8]	Holden & Freitas [1]
Number of ants	3000	3000	3000
Min. cases per rule	10	10	10
Max. uncovered cases	2	10	10
Rules for convergence	10	10	20

The parameters of the model used by previous authors and in this study are shown in Table 1. Each constructed rule had to cover a certain minimum number of cases in the training set. Each time a rule is produced the number of training set cases covered by that rule is found. If the number of uncovered cases is greater than the parameter maximum number of uncovered cases, then the rule is extended by adding another rule as an ELSE clause. The fourth parameter is the number of times an identical rule set must be produced by consecutive ants to produce convergence.

4. Results

The rule sets derived over the first five experimental runs were as follows, where the values in parentheses are the numbers of disputed papers which were classified at each stage of the rule's application.

```
IF (COMMONLY < 143 AND THERE < 2510 )
      THEN MADISON (11);
ELSE HAMILTON (1).


IF (PARTICULARLY < 189 AND WHILST < 116)
      THEN HAMILTON (2);
ELSE IF (ALSO < 462 AND TO > 37818)
      THEN HAMILTON (0);
ELSE IF (COMMONLY < 143)
      THEN MADISON (9);
UNCLASSIFIED (1).

IF (CONSEQUENTLY < 128 AND  ON < 4820)
      THEN HAMILTON (2);
ELSE IF (THERE < 2510 AND VIGOR < 122)
```

```
      THEN MADISON (9);
ELSE HAMILTON (1).

IF (ACCORDING < 1545 AND ALTHOUGH < 72 AND THOUGH < 2745 AND WHILST < 116)
      THEN HAMILTON (3);
ELSE IF (THERE > 2510)
      THEN HAMILTON (1);
ELSE MADISON (8).

IF (UPON > 2199)
      THEN HAMILTON (0);
ELSE IF (TO > 37818)
      THEN MADISON (10);
ELSE IF (INNOVATION < 562)
      THEN HAMILTON (2).
```

Over 10 experimental runs, each time using a different random seed, a total of 95 papers were assigned to Madison, 24 to Hamilton and 1 was unclassified. Assuming the disputed papers were written by Madison, the average success rate over the experimental runs was 79.1%. In the two experimental runs by Holmes and Forsyth's genetic algorithm, on one occasion all 12 disputed papers were assigned to Madison, while the other time 10 were assigned to Madison and 2 to Hamilton, giving an average success rate of 91.7%. They stated that the main problem with rule-based approaches to stylometry is the danger of misclassification, but the advantage is the ease of interpretation of rule sets containing just a few words compared with neural network weight matrices and numeric discriminant functions. Kjell used a large variety of neural network configurations, resulting in between 0 and 3 disputed papers being attributed to Hamilton.

Acknowledgement

Electronic versions of the Federalist Papers were kindly provided by Project Gutenburg at Illinois Benedictine College.

References

[1] N. Holden and A. A. Freitas (2004). Web Page Classification with an Ant Colony Algorithm. In: *PPSN VIII, 8th International Conference on Parallel Problem Solving from Nature*, Birmingham, UK, September 18-22. To appear in LNCS, Springer Verlag

[2] D. I. Holmes and R. S. Forsyth (1995). The Federalist Revisited: New Directions in Authorship Attribution. In: *Literary and Linguistic Computing* 10(2): pages 111-127.

[3] B. Kjell (1994). Authorship Determination using Letter Pair Frequency Features with Neural Network Classifiers. In: *Literary and Linguistic Computing* 9(2): pages 119-124.

[4] B. Liu, H. A. Abbass and B. McKay (2004). Classification Rule Discovery with Ant Colony Optimization. In *IEEE Computational Intelligence Bulletin*, 3(1): pages 31-35.

[5] W. B. McColly and D. Weier (1983). Literary Attribution and Likelihood Ratio Tests - the Case of the Middle English Pearl Poems. In: *Computers and the Humanities* 17: pages 65-75.

[6] T. V. N. Merriam (1989). An Experiment with the Federalist Papers. In: *Computers and the Humanities* 23: pages 251-254.

[7] F. Mosteller and D. L. Wallace (1984). *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*. Addison-Wesley: Reading, MA.

[8] R. S. Parpinelli, H. S. Lopes and A. A. Freitas (2002). Data Mining with an Ant Colony Optimization Algorithm. In: H. A. Abbass, R. A. Starker, C. S. Newton (Eds.) *Data Mining: a Heuristic Approach*, pages 192-208. London: Idea Group Publishing, 2002.

[9] F. J. Tweedie, S. Singh and D. I. Holmes (1996). Neural Network Applications in Stylometry: The Federalist Papers. In Computers and the Humanities 30(1): pages 1-10.