Notes on Authorship Attribution and Text Classification
(Richard Forsyth, December 2007)

Logically, authorship attribution is a kind of text classification, but it has some special features and is often approached by different people from different disciplines, so I propose to treat the two topics as separate but connected. In this document I consider two approaches to these related tasks: (1) the (pre-2002) Burrows approach to authorship attribution; (2) a Bayesian approach to classifying text segments, by author or other category scheme.

1. The "Classical" Burrows Approach

As Holmes (1994) has shown, a great variety of linguistic variables have been used in authorship studies. My own preference is to concentrate on variables which, in a sense, emerge from the texts under consideration.

A number of studies have appeared (e.g. Burrows, 1989, 1992; Binongo, 1994; Burrows & Craig, 1994; Holmes & Forsyth, 1995; Forsyth & Holmes, 1996; Tweedie et al., 1998; Forsyth et al., 1999) in which the features used as indicators are not imposed by the prior judgement of the analyst but are found by straightforward procedures from the texts under scrutiny. Such textual features have been used by Burrows (1992) as well as Binongo (1994), among others, not only in authorship attribution but also to distinguish among genres. This approach involves finding the most frequently used words and treating the rate of usage of each such word in a given text as a feature. The exact number of common words used varies by author and application. Burrows and colleagues (Burrows, 1992; Burrows & Craig, 1994) discuss examples using anywhere from the 50 to 100 most common words. Binongo (1994) uses the commonest 36 words (after excluding pronouns). Greenwood (1995) uses the commonest 32 (in New Testament Greek). Most such words are *function words*, and thus this approach can be said to continue the tradition, pioneered by Mosteller & Wallace (1964 / 1984), of using frequent function words as markers.

In fact, these studies (and some others) can be lumped together as applications of what may be called the "Burrows Approach", which is outlined below.

1. Pick the N most common words in the corpus under investigation. N may be from 15 to 100. (Manual preprocessing is sometimes done, e.g. distinguishing "that"-demonstrative from "that"-conj.)

2. Compute the occurrence rate of these N words in each text or text-unit, thus converting each text into an N-dimensional vector of numbers.

3. Apply techniques of multivariate data analysis to reveal patterns, especially:
   Principal Components Analysis;
   Cluster Analysis;
   Discriminant Analysis.

4. Interpret the results (with care!).

A striking success of this method is described by Burrows (1992) on prose works by the Bronte sisters. He took 4000-word samples of first-person fictional narrative from novels by the three sisters Anne, Charlotte and Emily, and was able to show that they fell into three distinct clusters. Given three such authors, linked by heredity and upbringing, writing in the same genre at around the same time, this was an impressive feat.

A number of studies have followed this approach, the majority of which have been on English-language texts. It should be said that John Burrows himself has developed other procedures for investigating authorship (Burrows, 2002; Burrows, 2006) but I don't feel competent to explain them in any depth. In any case, what I term the "classical" (pre-2002) Burrows approach still has plenty of mileage in it.

There is no definitive statement by Burrows (1992) or his successors on deciding exactly how many words to use. Generally about fifty are used, with the implication being that they should be among the most common in the language, and that content words should be avoided. As it is generally considered inadvisable to have more columns (features) than rows (texts) and as the demonstration dataset contains 36 texts, the examples shown here employ 36 words.

1.1 Data

The demonstration dataset consists of a selection of texts by Alexander Hamilton and James Madison, the two main authors of the *Federalist* papers, which were published in 1787-1788 and have never been out of print since. These essays gave rise to a celebrated and difficult case of disputed authorship which was subject to a ground-breaking stylometric analysis by Mosteller & Wallace (1984 [1964]) and which has become an accepted benchmark in the field of authorship attribution. Further details can be found in Holmes & Forsyth (1995). For the present investigation it should be noted that 31 **undisputed** papers by the two authors were chosen, 17 by Hamilton and 14 by Madison. In addition, two state of the union addresses given by Madison when he was president (in 1811 and 1813) were added to make the amount of text by both authors more nearly balanced. One jointly written paper (number 19) and two disputed papers (numbers 49 and 63) were also added to make 36 texts altogether.

1.2 Some Results

1.2.1 Turning Documents into Numeric Feature Vectors

The Burrows approach, like many others, relies on a pre-processing step in which texts (essentially variable-length strings of characters) are transformed into fixed-length vectors of numbers. In this case the numbers are relative frequencies of word-usage -- typically rates per 100 words, sometimes rates per 1000 words. This can be done with concordancing software such as Wordsmith/Tools, but I prefer to use a couple of programs of my own, written in Python, for this purpose. (These programs are in the public domain, and can be downloaded from my website: see final section on Websites at the end of this document for address.)

The first of these programs, voclist.py, creates an ordered word-frequency listing. Example output of voclist.py, when applied to the 36 texts in our Federalist corpus, follows.

```
    c:\mole\feds\hm36  Mon Nov 26 11:42:13 2007
the               8892    1  9.7370    9.7370  9.7646    9.7646  7.6135  9.4297 13.0435
of                5601    2  6.1332   15.8702  6.1311   15.8958  5.1684  6.0146  7.5695
to                3270    3  3.5807   19.4510  3.6388   19.5346  2.3375  3.4961  5.3422
and               2473    4  2.7080   22.1590  2.7056   22.2402  1.7829  2.6144  4.0347
in                2049    5  2.2437   24.4027  2.2318   24.4720  1.4469  2.1698  3.4518
a                 1853    6  2.0291   26.4317  2.0167   26.4887  0.9452  2.0049  2.7728
be                1741    7  1.9064   28.3382  1.9284   28.4171  0.6856  1.9014  3.2242
that              1264    8  1.3841   29.7223  1.3682   29.7853  0.6366  1.3148  2.2266
it                1219    9  1.3348   31.0571  1.3343   31.1196  0.6563  1.3158  2.3173
is                1097   10  1.2012   32.2584  1.1710   32.2906  0.2678  1.1074  2.1781
which             1038   11  1.1366   33.3950  1.1517   33.4423  0.6643  1.0653  1.8605
by                 903   12  0.9888   34.3838  0.9623   34.4045  0.4836  0.8372  1.8248
as                 812   13  0.8892   35.2730  0.8829   35.2875  0.1918  0.8800  1.2610
have               641   14  0.7019   35.9749  0.6967   35.9841  0.1146  0.6438  1.1536
this               640   15  0.7008   36.6757  0.6999   36.6841  0.2363  0.7233  1.1250
for                605   16  0.6625   37.3382  0.6619   37.3460  0.2557  0.6683  1.0631
not                590   17  0.6461   37.9843  0.6444   37.9904  0.1959  0.6376  0.9818
will               572   18  0.6264   38.6106  0.6356   38.6259  0.0490  0.5546  1.5302
on                 564   19  0.6176   39.2282  0.5940   39.2199  0.1113  0.5007  1.4747
with               545   20  0.5968   39.8250  0.6104   39.8303  0.2971  0.6125  1.0445
or                 544   21  0.5957   40.4207  0.5939   40.4242  0.2205  0.5356  1.2781
their              529   22  0.5793   41.0000  0.5813   41.0055  0.0365  0.5891  1.1612
would              518   23  0.5672   41.5672  0.6136   41.6191  0.1461  0.4429  1.7673
from               498   24  0.5453   42.1125  0.5447   42.1638  0.1751  0.5263  0.8841
an                 473   25  0.5179   42.6305  0.5327   42.6965  0.2293  0.5456  1.2585
are                472   26  0.5169   43.1473  0.5087   43.2053  0.1473  0.4257  1.0017
been               439   27  0.4807   43.6280  0.4711   43.6764  0.1823  0.4429  0.8859
they               427   28  0.4676   44.0956  0.4625   44.1389  0.0473  0.4407  0.9671
government         426   29  0.4665   44.5621  0.4713   44.6101  0.0979  0.4061  1.2610
states             404   30  0.4424   45.0045  0.4197   45.0299  0.0000  0.3388  1.3135
may                398   31  0.4358   45.4403  0.4224   45.4523  0.1390  0.3895  0.8915
its                348   32  0.3811   45.8214  0.3988   45.8510  0.1327  0.3605  0.8346
all                343   33  0.3756   46.1970  0.3837   46.2348  0.1334  0.3605  0.7673
but                340   34  0.3723   46.5693  0.3677   46.6024  0.0979  0.3668  0.8913
has                314   35  0.3438   46.9131  0.3342   46.9367  0.0496  0.3318  0.6759

upon               133   82  0.1456   57.1286  0.1546   57.1691  0.0000  0.0721  0.5348
```

The output is in 10 columns, as explained below.

1. Vocabulary item (normally an orthographic word).
2. Frequency of item in the texts processed.
3. Rank according to overall frequency (item 2, above).
4. Overall occurrence rate (all texts lumped together).
5. Cumulative occurrence rate.
6. Mean of mean occurrence rates (means calculated for individual files, then the mean of these means computed).
7. Cumulative occurrence rate using mean of means (item 6, above).
8. Lowest occurrence rate in all texts processed.
9. Median occurrence rate in all texts processed.
10. Highest occurrence rate in all texts processed.

Note: all occurrence rates are given as percentages.

This listing shows the top 35 words plus "upon", which actually ranked 82nd. Readers who know this problem will recognize tat "upon" is one of the most distinctive markers of Hamilton's authorship (in contrast with Madison's), so this inclusion offers us the option of "cheating" by drawing on pre-existing knowledge, to see how much difference that would make.

The second program, vocmole.py, takes a listing such as produced by voclist.py and uses the vocabulary items to produce a rectangular tab-delimited file suitable to be read into Excel, R or SPSS -- with rows being texts and columns being variables. (The first line of this file is a sequence of column names. These have V prefixed, and X suffixed if less than 3 characters, to avoid clashing with SPSS keywords.)

An extract follows to illustrate the format. Only the first four data lines are shown.

```
Name  ID      Location        Size    Vthe    VofX    VtoX    Vand    VinX    VaX     VbeX    Vthat
      VitX    VisX    Vwhich          VbyX    VasX    Vhave   Vthis   Vfor    Vnot    Vwill   VonX
      Vwith   VorX    Vtheir  Vwould          Vfrom   VanX    Vare    Vbeen   Vthey   Vgovernm
      Vstates Vmay    Vits    Vall    Vbut    Vhas    Vupon
fedpap08.txt  1       hm36    2049    7.6135  6.4910  3.8555  2.6354  2.0498  2.2450  1.7082  0.8785
      1.0249  1.0249  1.2689  0.5368  0.7809  0.8785  0.9273  0.4392  0.5857  0.5368  0.5368  0.6345
      0.8297  0.7809  1.3177  0.4392  0.6345  0.6833  0.7809  0.6345  0.1464  0.4880  0.3904  0.4880
      0.4392  0.4880  0.3904  0.1464
fedpap10.txt  2       hm36    2999    8.6362  5.1684  3.3011  4.0347  2.1007  2.6009  2.0340  1.0337
      1.5672  1.4005  1.3004  1.3004  0.6669  0.5335  0.3668  0.6002  0.4668  1.0003  0.6002  0.3668
      0.7336  0.7002  0.2001  0.4001  0.4668  0.9003  0.3001  0.3668  0.4335  0.1000  0.5335  0.3001
      0.1334  0.3668  0.1334  0.0000
fedpap12.txt  3       hm36    2159    8.1056  6.4382  3.7517  2.8717  2.5012  2.1769  1.8527  1.2506
      1.2043  1.0190  1.0653  0.6948  0.8800  0.6021  0.7874  0.3705  0.2779  0.5095  0.5558  0.8337
      0.3242  0.6484  1.0190  0.8800  0.5095  0.6948  0.3705  0.3705  0.3705  0.4632  0.1390  0.3705
      0.3242  0.3705  0.6021  0.3242
fedpap14.txt  4       hm36    2150    9.3023  5.6744  3.3023  2.7907  1.8605  1.7209  2.0930  1.5349
      1.4419  1.1628  1.8605  0.8372  1.1163  0.7907  0.6047  0.9302  0.6047  1.2093  0.7907  0.3256
      0.5116  0.8837  0.2326  0.5116  0.4186  0.3256  0.4186  0.7907  0.4186  0.5581  0.7442  0.3721
      0.2791  0.1860  0.3721  0.0000
....
```

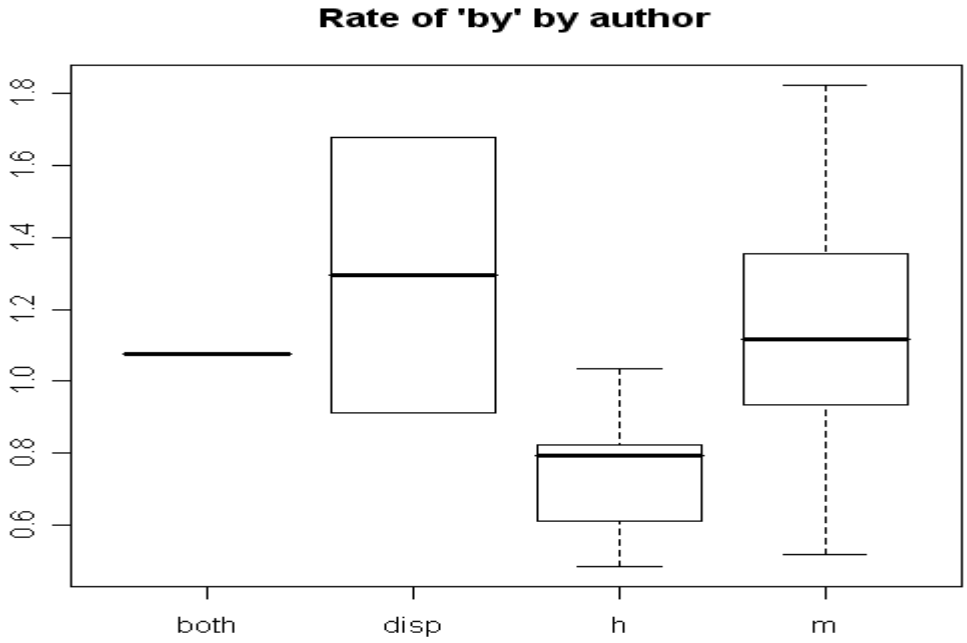
This data file can now be read into R (by the read.table function with header=T) for processing.

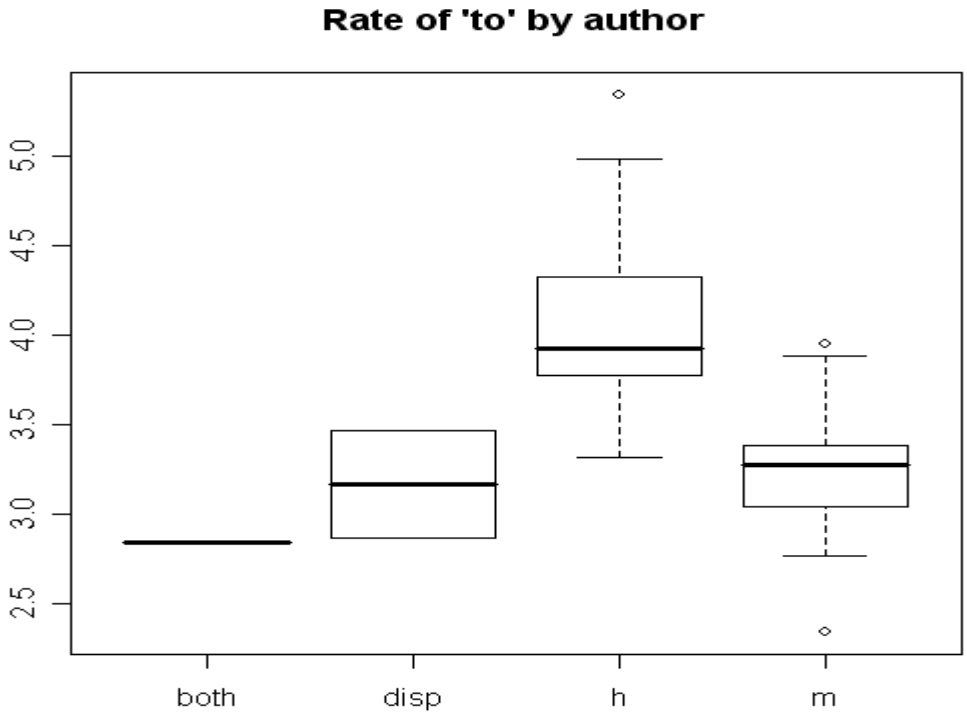
To assist in interpreting subsequent output, there follows a list linking authors and paper numbers for this 36-text Federalist corpus. (Disp=disputed; both=joint authorship.)

```
> fedlabs
 [1] "h 8"     "m 10"    "h 12"    "m 14"    "h 15"    "h 16"    "h 17"
 [8] "both 19" "h 23"    "h 33"    "m 37"    "m 38"    "m 39"    "m 40"
[15] "m 41"    "m 42"    "m 43"    "m 44"    "m 45"    "m 46"    "m 47"
[22] "m 48"    "disp 49" "h 59"    "disp 63" "h 65"    "h 66"    "h 68"
[29] "h 70"    "h 72"    "h 75"    "h 78"    "h 81"    "h 84"    "m 1811"
[36] "m 1813"
```

1.2.2 Univariate Exploration

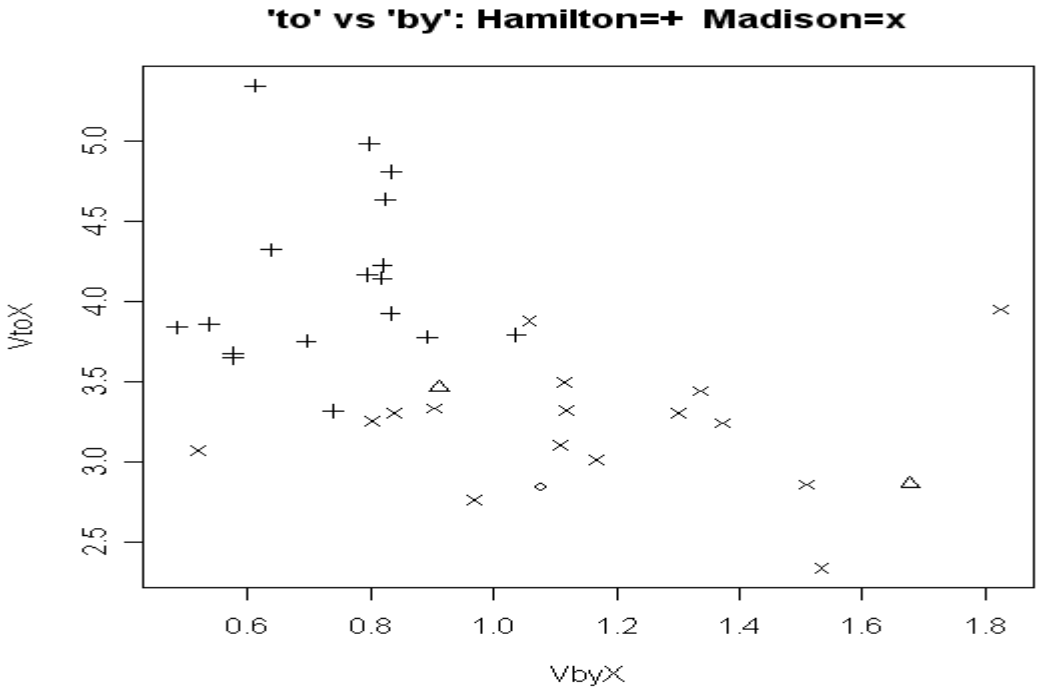
Some frequent function words that could serve as potential **markers**.

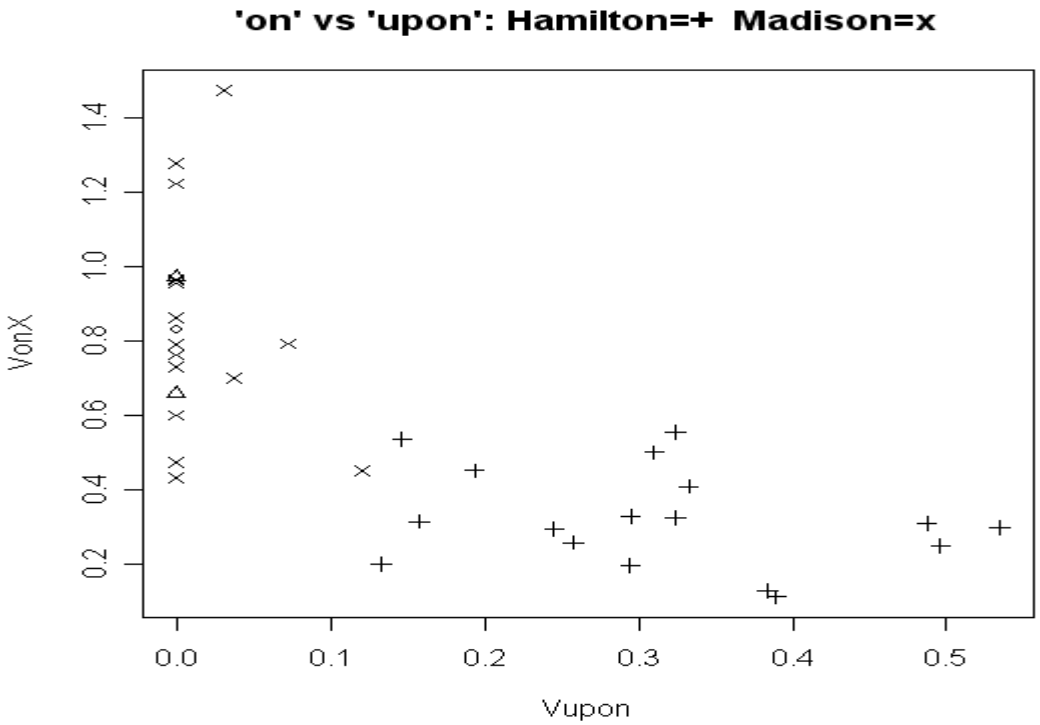




1.2.3 Bivariate Exploration

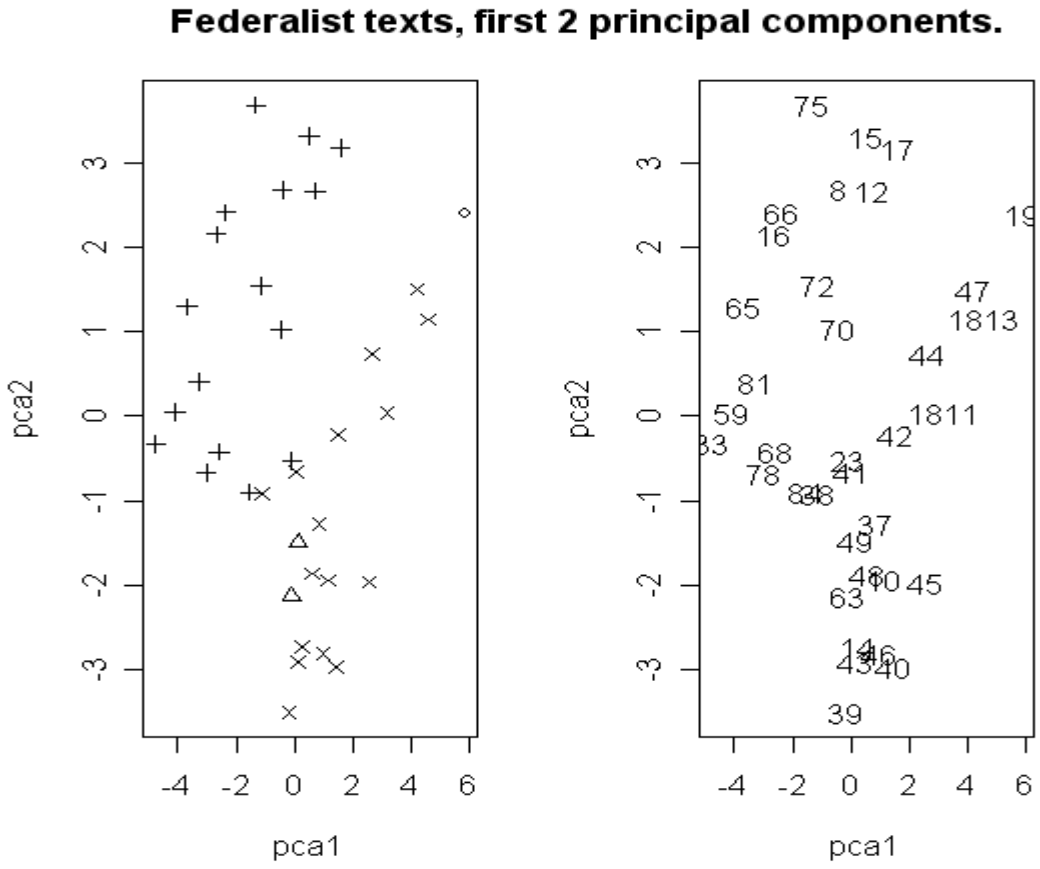
Often it is revealing to look at a 2-dimensional plot.





1.2.4 Multivariate Exploration: Principal Components Analysis (PCA)

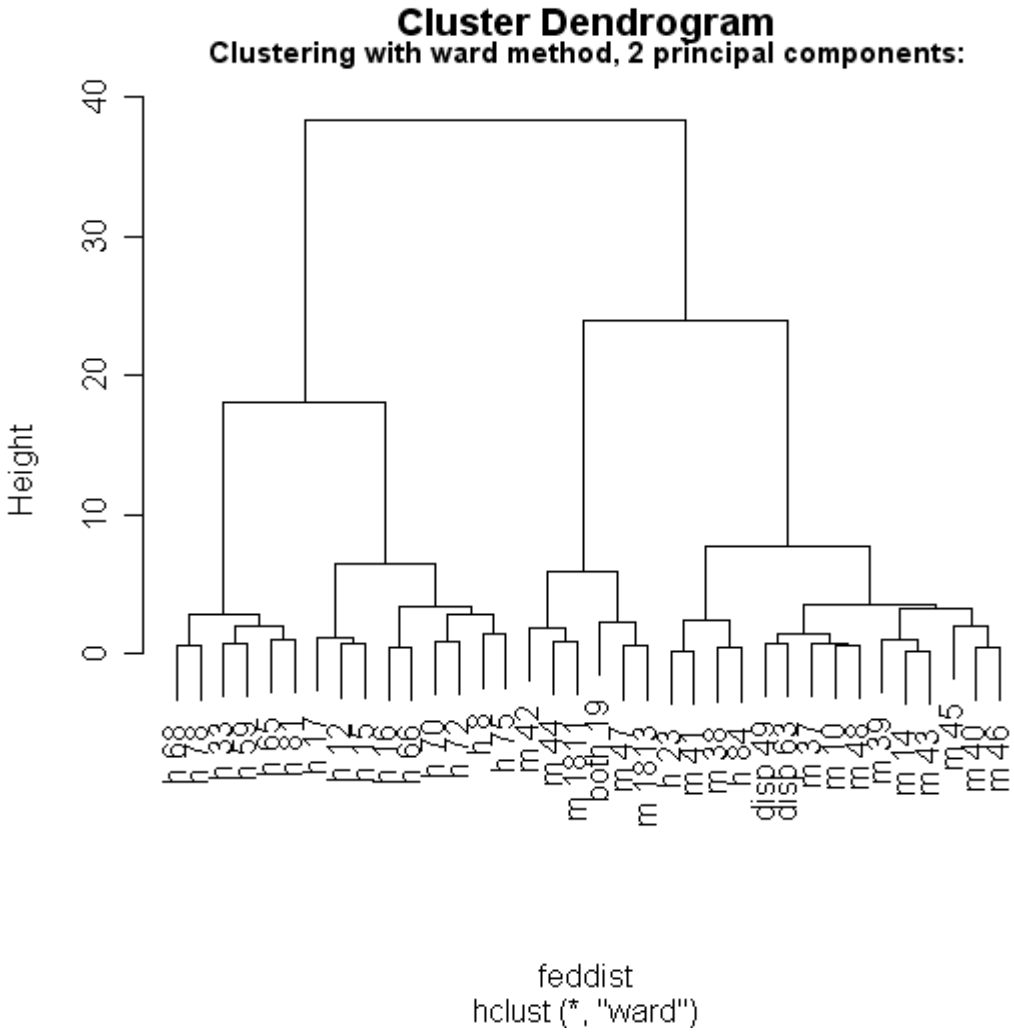
Principal components analysis is a data-reduction technique which aims to replace an original set of n variables by a derived set of uncorrelated variables, each of which is a linear combination of the original variables. The derived variables (components) are ordered so that the first accounts for more of the variance in the original data than the second and so on. If the original variables are correlated, it should be possible to account for most of the variation in the original variables by the first p components, where p is considerably less than n.



Here the first 2 components (from 36 original variables) account for 27% of the total variance. They also seem to give reasonably good separation by author. The disputed and joint samples fall within Madisonian territory on the graphs.

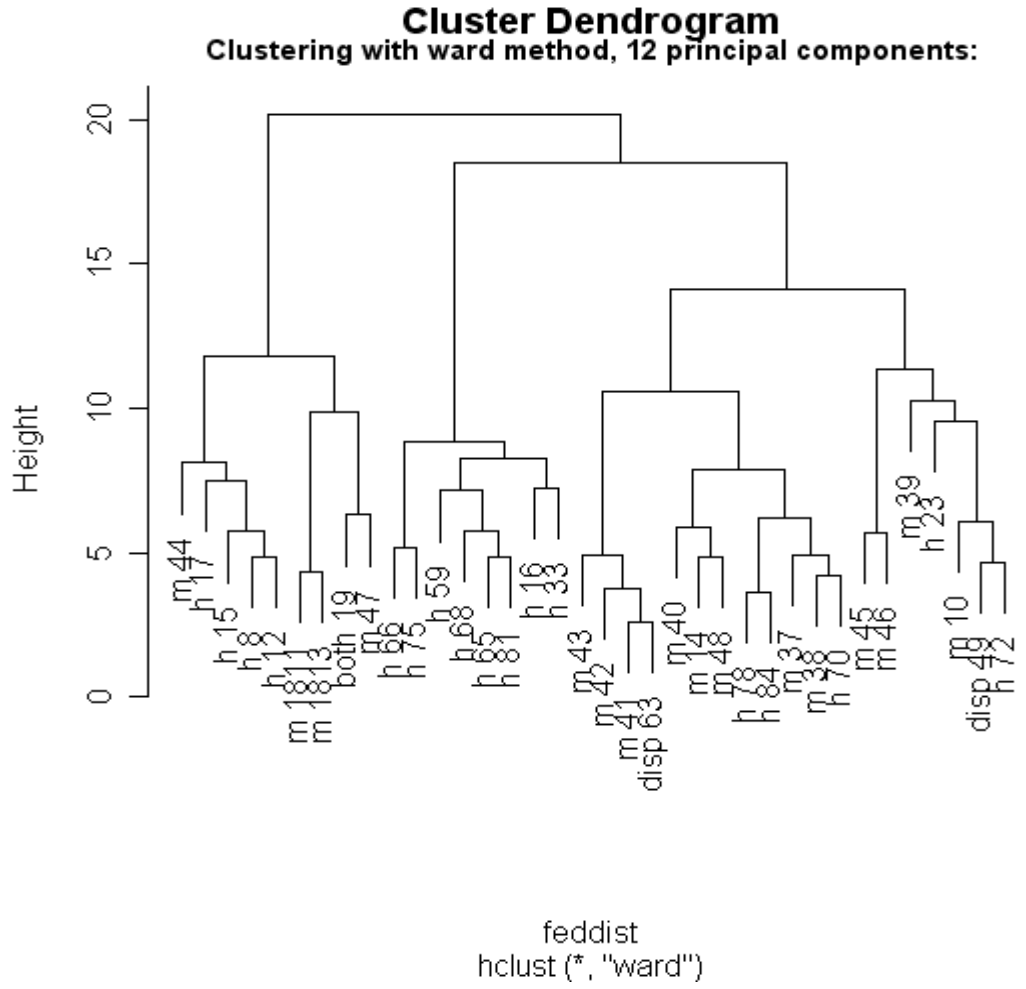
1.2.5 Multivariate Exploration: Hierarchical Clustering (with PC scores)

Cluster analysis is a technique designed to discover groupings within a dataset. In the present case, the first few principal components are used to compute inter-item distances, on which the clustering is based. (Here Ward's method (see Everitt, 1993) is used, as I think it has a sound rationale.)



Cluster analysis is an exploratory technique: there is no "right answer". How many components to use? There is no definitive answer to that. More doesn't always mean better.

In this dataset, the first 12 components account for 82% of the variance.



1.2.6 Multivariate Classification: Linear Discriminant Analysis (LDA), with PC scores

Discriminant analysis is method of allocating objects to classes based on the value of a mathematical formula. In LDA, the formula is linear, i.e. a weighted sum of feature values. Unlike cluster analysis, the true classes should be known for all objects in the *training data* in order to optimize the coefficients of the (linear) discriminant function. The output below was generated by a script in the R language, using the 33 undisputed texts as training cases and the three others as test cases. Using the first 2 principal components it makes 2 mistakes (using *cross-validation*) on the training data; and assigns all 3 test cases to Madison.

```
> fedslda(pc42,2,fedframe)

Cross-validated confusion matrix :
      gr
       both disp  h  m
  both    0    0  0  0
  disp    0    0  0  0
  h       0    0 16  1
  m       0    0  1 15
```

```
Classification errors :
8 4 3
8 0.0957428 0.9042572
11 3 4
11 0.6406403 0.3593597

Results on test data :
$class
[1] m m m
Levels: both disp h m

$posterior
              h         m
8  0.0001548181 0.9998452
23 0.0082386461 0.9917614
25 0.0026704456 0.9973296

$x
        LD1
8  2.325882
23 1.303937
25 1.594470

Call:
lda(pcax$scores[trainers, 1:dims], grouping = gr)

Prior probabilities of groups:
        h         m
0.5151515 0.4848485

Group means:
     Comp.1    Comp.2
h -1.686967  1.270172
m  1.425831 -1.273549

Coefficients of linear discriminants:
              LD1
Comp.1  0.6684626
Comp.2 -0.7139742
Warning messages:
1: groups both disp are empty in: lda.default(x, grouping, ...)
2: groups both disp are empty in: lda.default(x, grouping, ...)
>
```

Using the first 12 components the system makes no mistakes in cross-validation. It also assigns all 3 test cases to Madison.

```
> fedslda(pc42,12,fedframe)
Cross-validated confusion matrix :
      gr
       both disp  h  m
  both    0    0  0  0
  disp    0    0  0  0
  h       0    0 17  0
  m       0    0  0 16

Classification errors :

Results on test data :
$class
[1] m m m
Levels: both disp h m
```

```
$posterior
              h         m
8  5.963556e-05 0.9999404
23 4.403897e-06 0.9999956
25 1.711078e-09 1.0000000

$x
        LD1
8  1.702911
23 2.131768
25 3.424210
```

2. A Bayesian Approach

Thus what I term the "classical" Burrows approach still has plenty to offer in terms of both insight into the data and accuracy of categorization, if this challenging authorship problem can be taken as a benchmark (as it often has been). So why develop alternative methods?

1. The world doesn't stand still. (Burrows himself hasn't rested on his laurels.)
2. The Burrows approach uses only word frequencies -- not, for example, word transition rates, which tap into sequential information (and sequence is one of the fundamentals of language).
3. I needed to categorize quite short text segments. (The Burrow approach is usually used on blocks of several thousand words, and breaks down with chunks of less than 500 words.)
4. I'd like to skip the pre-processing stage in which text is turned into numeric vectors, and work directly with textual data.

2.1 In Praise of Paragraphs

In what follows the unit of analysis is the paragraph: the classification program's task is to classify individual paragraphs according to their author, not whole documents. Clearly classifying paragraphs (mean size 144 words) is a more challenging task than classifying entire documents (mean size over 2500 words).

2.2 Classification Algorithm

A large number of algorithms has been used for text classification (e.g. Yang, 1999; Stamatatos et al., 2001; Sebastiani, 2002; Peng et al., 2003). The algorithm used in the present investigation is essentially a generalization of that described in Khmelev & Tweedie (2001), which has been shown to give good results in the area of authorship attribution in both English and Russian. This algorithm -- which itself is a variant of the widely-used Naive Bayes Classifier, as described, for instance, in Mitchell (1997) -- creates a simple Markovian model of the language in the training dataset and uses Bayesian inference to arrive at probabilistic category assignments (on training or test data). I call it a *Bayes-Markov Classifier* (BMC).

The advantages of this simple and robust algorithm include the following: it requires no pre-processing step to select features (in effect, all features are used); it requires no external support software, such as taggers, or lexicons; and it could potentially be applied to languages other than English (though the present trial is on English texts). Moreover, it employs a Bayesian inferential

framework, which has served as the basis for several practical text-categorization systems, such as spam filtering (Sahami et al., 1998). However, I do not wish to claim that this algorithm is the best possible for this purpose, only that it achieves acceptable accuracy levels.

The system is as described by Khmelev & Tweedie (2001) with two extensions:

1) their system used character bigrams (pairs) as the basis for its language model, whereas the BMC permits n-grams of any length and allows word-based as well as character-based n-grams, and is thus more flexible;
2) their system simply ignored attributes with a zero frequency in the training data, whereas the BMC uses the so-called "m-estimate" procedure (see, for instance, Cestnik & Bratko, 1991) which has the side-effect of attenuating extreme probabilities, including zero and one; hence no attributes are completely ignored.

The algorithm is also very similar to that of Peng et al. (2003), the only differences being that BMC uses a different smoothing technique (item (2) above) and that it allows words as well as characters to be the basic units.

2.3 Codelearner

The Codelearner suite is a collection of software modules, written in Python, which assist in the task of categorizing short segments of text. The two most important programs are codelearner1.py and codelearner2.py which both implement the BMC algorithm described above. These use a very simple input format in which every text line that doesn't start with a coding prefix ("[[" by default) is treated as a text segment to be categorized, while lines that begin with the coding prefix are treated as annotation lines which assign values to variables. Annotation lines apply to the text line preceding them. The extract below, from Federalist paper 70, illustrates this format. Note that these text lines have been word-wrapped in the present document, but as far as the Codelearner software is concerned they consist of four text lines and four annotation lines. (Blank lines are ignored.) In effect each paragraph of the original essay becomes a line of text.

```
There can be no need, however, to multiply arguments or examples on this head. A feeble
Executive implies a feeble execution of the government. A feeble execution is but
another phrase for a bad execution; and a government ill executed, whatever it may be
in theory, must be, in practice, a bad government.
[[ by=Alexander Hamilton

Taking it for granted, therefore, that all men of sense will agree in the necessity of
an energetic Executive, it will only remain to inquire, what are the ingredients which
constitute this energy? How far can they be combined with those other ingredients which
constitute safety in the republican sense? And how far does this combination
characterize the plan which has been reported by the convention?
[[ by=Alexander Hamilton

The ingredients which constitute energy in the Executive are, first, unity; secondly,
duration; thirdly, an adequate provision for its support; fourthly, competent powers.
[[ by=Alexander Hamilton

The ingredients which constitute safety in the republican sense are, first, a due
dependence on the people, secondly, a due responsibility.
[[ by=Alexander Hamilton
```

In this instance, the variable of interest is "by" which identifies the author.

When codelearner1 or codelearner2 is run, it treats all lines annotated with the outcome variable as training data and all lines not annotated with the outcome variable as test data. It forms a Markov model from the training data and uses this to assign category codes probabilistically to the unannotated test data.

For the present experiment, author-coding annotations were removed from six of the 36 files, namely: 19 both, 33 h, 42 m, 49 disp, 63 disp, 1811 m.

The output format can be illustrated by the following extract (the first five segments/paragraphs from Federalist paper 42, a Madisonian essay).

```
THE second class of powers, lodged in the general government, consists of those which
regulate the intercourse with foreign nations, to wit: to make treaties; to send and
receive ambassadors, other public ministers, and consuls; to define and punish piracies
and felonies committed on the high seas, and offenses against the law of nations; to
regulate foreign commerce, including a power to prohibit, after the year 1808, the
importation of slaves, and to lay an intermediate duty of ten dollars per head, as a
discouragement to such importations.
[[ by=James Madison
[[ id=0
[[ probvec= 0.0000000 1.0000000

This class of powers forms an obvious and essential branch of the federal
administration. If we are to be one nation in any respect, it clearly ought to be in
respect to other nations.
[[ by=James Madison
[[ id=1
[[ probvec= 0.0010596 0.9989404

The powers to make treaties and to send and receive ambassadors, speak their own
propriety. Both of them are comprised in the articles of Confederation, with this
difference only, that the former is disembarrassed, by the plan of the convention, of
an exception, under which treaties might be substantially frustrated by regulations of
the States; and that a power of appointing and receiving "other public ministers and
consuls," is expressly and very properly added to the former provision concerning
ambassadors. The term ambassador, if taken strictly, as seems to be required by the
second of the articles of Confederation, comprehends the highest grade only of public
ministers, and excludes the grades which the United States will be most likely to
prefer, where foreign embassies may be necessary. And under no latitude of construction
will the term comprehend consuls. Yet it has been found expedient, and has been the
practice of Congress, to employ the inferior grades of public ministers, and to send
and receive consuls.
[[ by=Alexander Hamilton
[[ id=2
[[ probvec= 0.9994431 0.0005569

It is true, that where treaties of commerce stipulate for the mutual appointment of
consuls, whose functions are connected with commerce, the admission of foreign consuls
may fall within the power of making commercial treaties; and that where no such
treaties exist, the mission of American consuls into foreign countries may perhaps be
covered under the authority, given by the ninth article of the Confederation, to
appoint all such civil officers as may be necessary for managing the general affairs of
the United States. But the admission of consuls into the United States, where no
previous treaty has stipulated it, seems to have been nowhere provided for. A supply of
the omission is one of the lesser instances in which the convention have improved on
```

```
the model before them. But the most minute provisions become important when they tend
to obviate the necessity or the pretext for gradual and unobserved usurpations of
power. A list of the cases in which Congress have been betrayed, or forced by the
defects of the Confederation, into violations of their chartered authorities, would not
a little surprise those who have paid no attention to the subject; and would be no
inconsiderable argument in favor of the new Constitution, which seems to have provided
no less studiously for the lesser, than the more obvious and striking defects of the
old.
[[ by=James Madison
[[ id=3
[[ probvec= 0.0000000 1.0000000

The power to define and punish piracies and felonies committed on the high seas, and
offenses against the law of nations, belongs with equal propriety to the general
government, and is a still greater improvement on the articles of Confederation. These
articles contain no provision for the case of offenses against the law of nations; and
consequently leave it in the power of any indiscreet member to embroil the Confederacy
with foreign nations. The provision of the federal articles on the subject of piracies
and felonies extends no further than to the establishment of courts for the trial of
these offenses. The definition of piracies might, perhaps, without inconveniency, be
left to the law of nations; though a legislative definition of them is found in most
municipal codes. A definition of felonies on the high seas is evidently requisite.
Felony is a term of loose signification, even in the common law of England; and of
various import in the statute law of that kingdom. But neither the common nor the
statute law of that, or of any other nation, ought to be a standard for the proceedings
of this, unless previously made its own by legislative adoption. The meaning of the
term, as defined in the codes of the several States, would be as impracticable as the
former would be a dishonorable and illegitimate guide. It is not precisely the same in
any two of the States; and varies in each with every revision of its criminal laws. For
the sake of certainty and uniformity, therefore, the power of defining felonies in this
case was in every respect necessary and proper.
[[ by=James Madison
[[ id=4
[[ probvec= 0.0000000 1.0000000
```

The lines starting with "[[probvec=" give the program's probability estimates for each of the outcome categories (2 in this case, with Hamilton first and Madison second) relating to the preceding text segment (paragraph). Note that the program has made an error on the third paragraph (id=2) which it assigned to Hamilton.

In fact the program, using repeated subsampling (a form of cross-validation) on the segments with known categories, estimates its accuracy rate as 85.5%, i.e. over 85% of the paragraphs are assigned to their true author. In this case where author identification for whole essays has been regarded as a difficult problem, this seems quite a promising result.

For reference, an output listing from codelearner1.py is reproduced below.

```
C:\CL07\pystuff\CodeLearner1.py started on Fri Nov 30 12:57:04 2007

Parameter settings after reading C:\CL07\pystuff\fedword.pf:
('atomize', 1)
('dumpfile', 'c:\\cl07\\feds\\outpath\\fedumpw.txt')
('filetype', '.txt')
('foldcase', 1)
('gramsize', 1)
('gramunit', 'word')
('kappaval', 0)
('missing', '~')
('name', 'C:\\CL07\\pystuff\\fedword.pf')
```

```
('outfile', 'c:\\cl07\\feds\\outpath\\fedword.txt')
('outpath', 'c:\\cl07\\feds\\outlines')
('precode', '[[')
('prepath', 'c:\\cl07\\feds\\fedlines')
('segname', 'by')
('segtran', {})
('skiptest', 0)
('stopfile', 'c:\\dict\\cobuild.111')
('textpath', 'c:\\cl07\\feds\\fedlines')
('topsize', '800')
('wordonly', 0)
20 parameter values set.

Input text folder : c:\cl07\feds\fedlines
fedpap08.txt                   15
fedpap10.txt                   23
fedpap12.txt                   13
fedpap14.txt                   12
       [.... 30 lines omitted to save space ....]
sou1811.txt                    27
sou1813.txt                    38

Total number of segments =    631
Number of forecasts made =    103

Pre-testing on 528 instances with known category codes:
Category-code variable : by
Training-set size =    495, holdout sample size =   33.

Gram-size =  0; trials = 1056; scores =  855,  0.8097
Gram-size =  1; trials = 1056; scores =  903,  0.8551
Gram-size =  2; trials = 1056; scores =  899,  0.8513
Gram-size =  3; trials = 1056; scores =  792,  0.7500

Results for gramsize =  1 :

Confusion matrix: rows=predicted, cols=true categories.

*      Alexander Hamilton  James Madison
Alexander Hamilton  432    97
James Madison       56     471

Confusion matrix: predicted+true+freq, SPSS-readable format.

Alexander Hamilton  Alexander Hamilton  432
Alexander Hamilton  James Madison       97
James Madison Alexander Hamilton        56
James Madison James Madison             471
```

The following table gives the paragraph assignments for the six (uncoded) test files.

Text	**Paragraphs assigned to AH**	**Paragraphs assigned to JM**
19 both	7	13
33 h	7	1
42 m	3	17
49 disp	1	6
63 disp	5	16
1811 m	3	24

For the three undisputed files the majority is clearly in favour of the true author. For the disputed papers the majority clearly favours Madison. The interesting case is number 19, a joint paper. Scholars have generally leaned towards Madison as the primary author, with Hamilton's role seen as merely supplying some notes. However, this result raises the possibility of a more integrated collaboration, with each author primarily responsible for different chunks of the whole.

Though this conclusion is somewhat speculative, it does highlight the fact that systems that categorize entire documents cannot even address such a question, thus emphasizing the value of text categorization systems that work with smaller segments of text.

References

Binongo, J.N.G. (1994). Joaquin's Joaquinesquerie, Joaquinesquerie's Joaquin: A Statistical Expression of a Filipino Writer's Style. *Literary & Linguistic Computing*, 9(4), 267-279.

Burrows, J.F. (1989). 'An Ocean Where each Kind...': Statistical Analysis and Some Major Determinants of Literary Style. *Computers & the Humanities*, 23, 309-321.

Burrows, J.F. (1992). Not unless you Ask Nicely: the Interpretive Nexus between Analysis and Information. *Literary & Linguistic Computing*, 7(2), 91-109.

Burrows, J.F. (2002). 'Delta': a measure of stylistic difference and a guide to likely authorship. *Literary & Linguistic Computing*, 17(3), 267-287.

Burrows, J.F. (2006). All the way through: testing for authorship in different frequency strata. *Literary & Linguistic Computing*, 21(1), 1-21.

Burrows, J.F. & Craig, D.H. (1994). Lyrical Drama and the "Turbid Montebanks": Styles of Dialogue in Romantic and Renaissance Tragedy. *Computers & the Humanities*, 28, 63-86.

Cestnik, B. & Bratko, I. (1991). On estimating probabilities in tree pruning. *Fifth European Working Session on Learning*, EWSL 91, Porto, Portugal, Springer-Verlag.

Everitt, B.S. (1993). *Cluster Analysis*, third edition. London: Arnold.

Forsyth, R.S. & Holmes, D.I. (1996). Feature-Finding for Text Classification. *Literary & Linguistic Computing*, 11(4), 163-174.

Forsyth, R.S., Holmes, D.I. & Tse, E.K. (1999). Cicero, Sigonio and Burrows: investigating the authenticity of the "Consolatio". *Literary & Linguistic Computing*, 14(3), 375-397.

Greenwood, H.H. (1995). Common Word Frequencies and Authorship in Luke's Gospel and Acts. *Literary & Linguistic Computing*, 10(3), 183-187.

Hamilton, A., Madison, J. & Jay, J. (1992 [1788]). *The Federalist Papers*. London: Dent. (ed.) W.R. Brock.

Holmes, D.I. (1994). Authorship Attribution. *Computers & the Humanities*, 28, 1-20.

Holmes, D.I. & Forsyth, R.S. (1995). The "Federalist" Revisited: New Directions in Authorship Attribution. *Literary & Linguistic Computing*, 10(2), 111-127.

Khmelev, D.V. & Tweedie, F.J. (2001). Using Markov chains for identification of writers. *Literary & Linguistic Computing*, 16(3), 299-307.

Mitchell, T.M. (1997). *Machine Learning*. New York: McGraw-Hill.

Mosteller, F. & Wallace, D.L. (1984). *Applied Bayesian and Classical Inference: the Case of the Federalist papers.* Springer-Verlag, New York. [First edition: 1964.]

Peng, F., Schuurmans, D., Keselj, V. & Wang, S.. (2003). Language independent authorship attribution using character level language models. *European Association of Computational Linguistics*, EACL2003, Budapest, 12-17 April, 2003.

Sahami, M., Dumais, S., Heckerman, D. & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *AAAI98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, 27 July 1998.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47.

Stamatatos, E., Fakotakis, N. & Kokkinakis, G, (2001). Automatic text categorization in terms of genre and author. *Computational Linguistics*, 26(4), 471-495.

Tweedie, F.J., Holmes, D.I. & Corns, T.N. (1998). The Provenance of *De Doctrina Christiana*, Attributed to John Milton: A Statistical Investigation. *Literary & Linguistic Computing*, 13(2), 77-87.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1, 69-90.

Websites

Federalist papers
http://www.yale.edu/lawweb/avalon/federal/fed.htm

Python
http://www.python.org

R-project
http://www.r-project.org/

RF home page (for the moment)
http://www.psychology.nottingham.ac.uk/staff/rsf