Stylometry Using Adjacent Word Graphs

Leon Maurer

Math 76

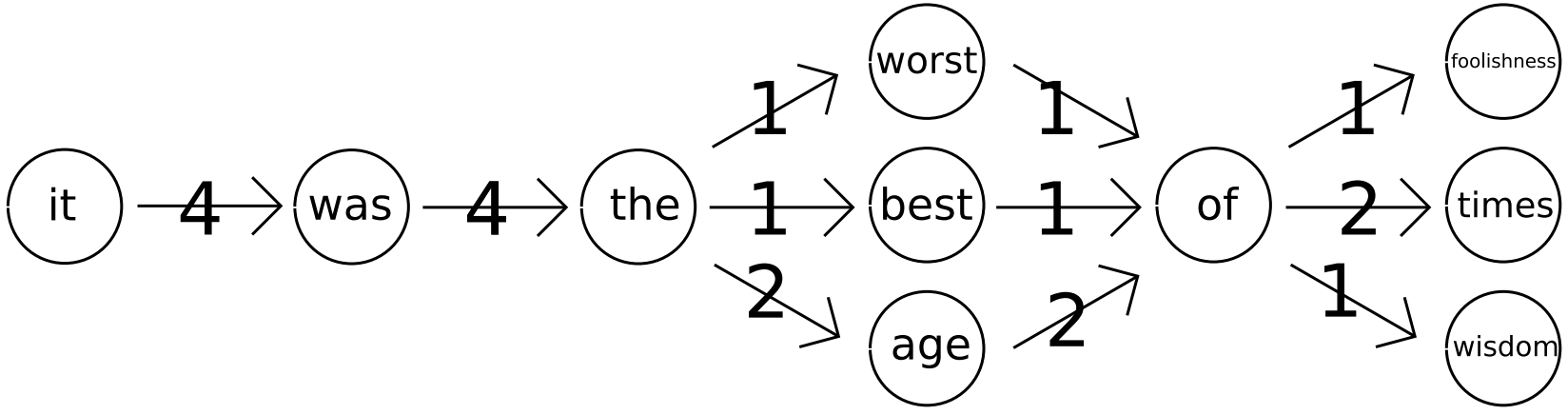
The plan

1. Take works, chop them up, and make graphs out of them
2. Perform HITS on graphs and find Hub vectors
3. Do Principle Component Analysis on the vectors
4. Squint at results
5. ???
6. Profit!!!

Making the Graphs

- Words are vertices
- Directed edges from one word to the next
- If the edge already exists, add one to its weight
- Restart at punctuation

It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness...



HITS

Review of HITS algorithm:

1. Start with $\vec{h_0} = (1, 1, 1...)$
2. $\vec{a_{t+1}} = A^T \vec{h_t}$
3. $\vec{h_{t+1}} = A\vec{a_{t+1}}$
4. Repeat until $\vec{h_{t+1}} \approx \vec{h_t}$ when normalized

Where A is the adjacency matrix, $\vec{h}$ is the hub vector, and $\vec{a}$ is the authority vector.

For these graphs, $\vec{h}$ converged quickly. Typically $\vec{h_3} \cdot \vec{h_2} > .99$

Typical $\vec{h}$s

of	.6452	.6039	.7557	.6147	.5735
in	.5161	.5286	.3995	.4389	.5259
and	.2968	.3497	.2463	.3352	.2506
to	.2326	.2333	.2118	.2980	.2504
on	.1613	.2110	.1932	.2220	.2254
at	.1592	.0886	.0555	.1581	.2296
with	.1181	.0916	.0775	.1460	.1921
for	.0570	.1626	.1198	.1136	.1483
from	.0883	.0927	.1189	.1413	.1234
by	.1385	.0846	.1031	.1047	.0828
was	.0622	.0966	.0646	.1450	.1150
through	.1360	.0533	.0423	.0563	.0536

Typical $\vec{a}$s

the	.9543	.8981	.9231	.9117	.8628
a	.1657	.3598	.3016	.2895	.3176
his	.0055	.0627	.0367	.1643	.2719
it	.0636	.1138	.0813	.0786	.1299
that	.0649	.0555	.0335	.0489	.0484
this	.0380	.0450	.0545	.0534	.0376
be	.0840	.0178	.0175	.0540	.0347
them	.0179	.0572	.0686	.0365	.0259
one	.0678	.0197	.0399	.0280	.0425
all	.0338	.0665	.0444	.0252	.0265
her	.0000	.0254	.0181	.0318	.1113
their	.0148	.0415	.0450	.0301	.0382

Principle Component Analysis

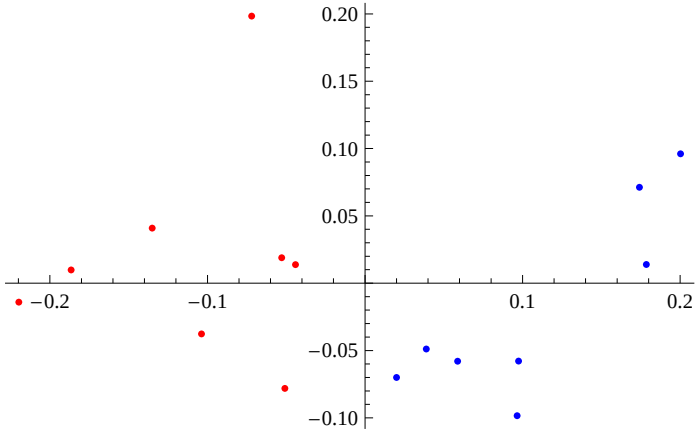
Simply taking the dot product of the $\vec{h}$ vectors doesn't reveal much about authorship – the dot products all are $\approx .95$. So it's time to do PCA.

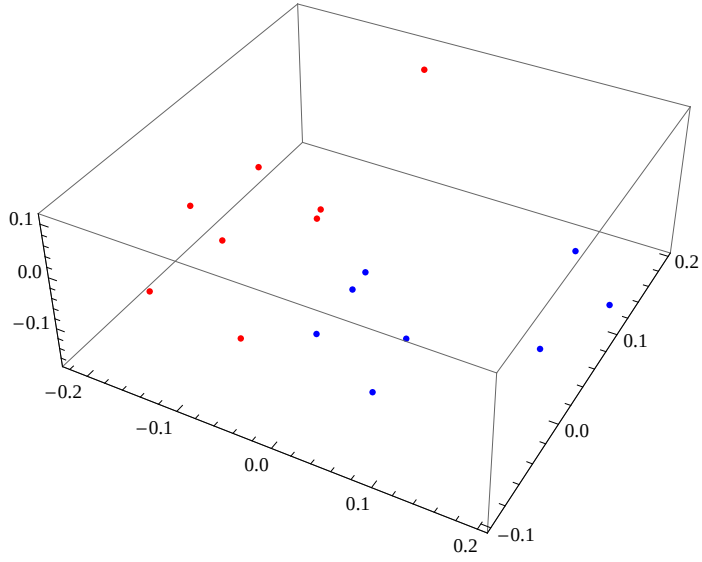
- Each $\vec{h}$ has thousands of entries – it's too big
- Cut all $\vec{h}$s down to the ≈ 30 words with the highest average values
- The sum of the top 2 or 3 eigenvalues is often about half of the total, so 2 or 3 dimensions should provide an ok representation

Twain vs. Dickens

Red dots are from Innocents Abroad. Blue dots are from A Tale of Two Cities.

- Works chosen because they are quite different – if this method works, it will work here
- 8 chunks of 4000-6000 words from each book

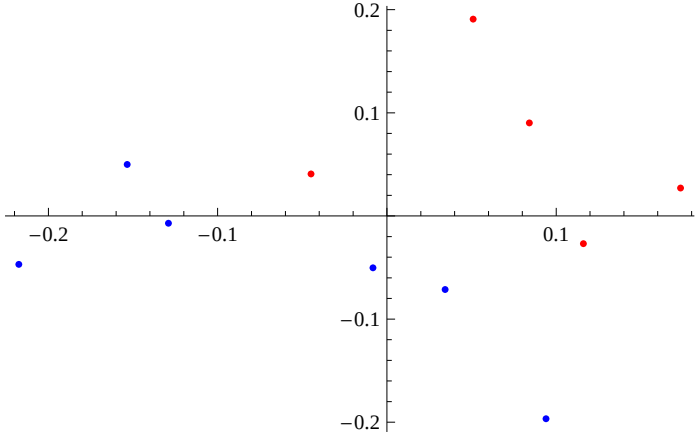


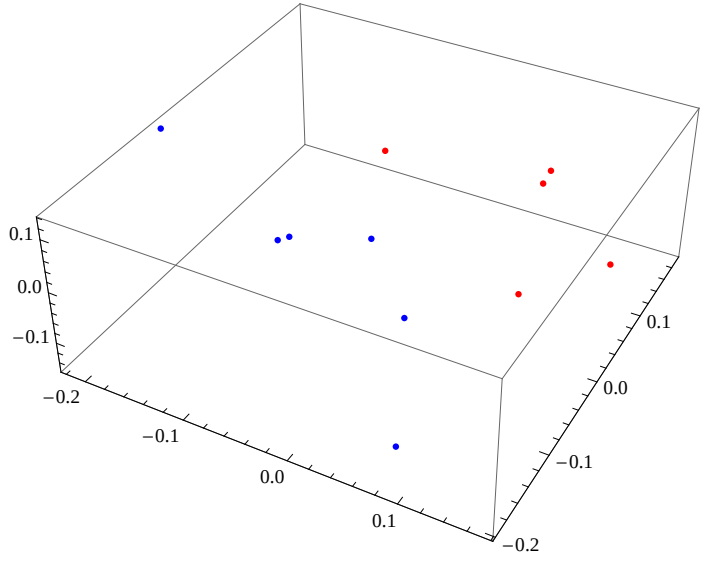


Eliot vs. Gaskell

Red dots are from Middlemarch. Blue dots are from North and South.

- Works chosen because they are similar (I am told) – both written by women in Victorian England and have some themes in common
- Chunks are again 4000-6000 words

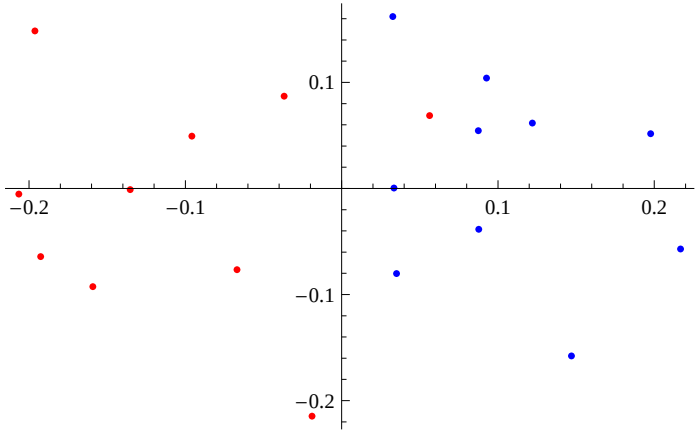


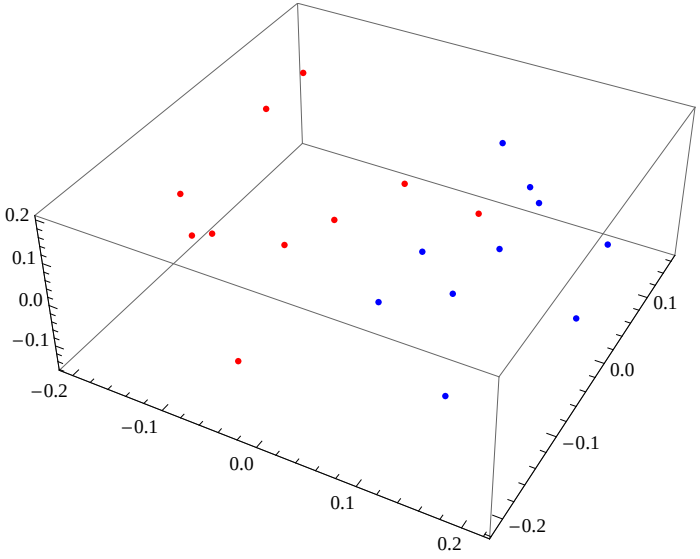


Darwin vs. Spencer

Red dots are from The Descent of Man. Blue dots are from Essays on Education and Kindred Subjects.

- Wanted to test some non-fiction works.
- Chunks are somewhat larger





Closing Thoughts

The method shows some promise. What might improve it?

- To some extent, bigger chunks are better. I could do whole books at once if I had more RAM.
- Program can probably be tweaked for some more speed.
- Is it a good thing that a few words have very high scores? If not, we could re-weight the edges non-linearly.
- Make use of both $\vec{a}$ and $\vec{h}$
- Remove the squint step and do clustering in higher dimensions instead.

References

- George Bebis, Principal Components Analysis, http://www.cse.unr.edu/ be-bis/MathMethods/PCA/lecture.pdf
- Jon M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, http://www.cs.cornell.edu/home/kleinber/auth.pdf
- M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, http://arxiv.org/abs/physics/0605087
- Andrew Y. Ng, Alice X. Zheng and Michael Jordan, Link analysis, eigenvectors, and stability, http://ai.stanford.edu/ ang/papers/ijcai01-linkanalysis.pdf