matpalm.com/lsa-via-svd/index.html

latent semantic analysis via the singular value decomposition

following on from some previous work on classifying documents i wanted to see how well latent semantic analysis (lsa) does at classifying documents.

wha?

usually when comparing documents we do so using the fundamental unit of the text; the actual terms themselves.
lsa gives a way of comparing documents at a higher level than the terms by introducting a concept called the *feature*.
the singular value decomposition (svd) is a way of extracting features from documents.

an example

lets go through a high level example to help build the initution and see what these features 'look like'

first let's introduce the *term occurance matrix*, a common way to describe a corpus, where rows represent terms and columns represent documents.
the value of matrix element $a_{i,j}$ denotes that the i^{th} term occured n times in the j^{th} document.

consider the, extremely contrived, documents...

```
d1: modem the steering linux. modem, linux the modem. steering the modem. linux!
d2: linux; the linux. the linux modem linux. the modem, clutch the modem. petrol.
d3: petrol! clutch the steering, steering, linux. the steering clutch petrol. clutch the petrol; the clutch.
d4: the the the. clutch clutch clutch! steering petrol; steering petrol petrol; steering petrol!!!!
```

which is represented as the 6x4 document term matrix below (colour introduced just to help see patterns)

	d1	d2	d3	d4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4

straight away we can see that, based on what words the documents contain, that doc1 and doc2 are alike and doc3 and doc4 are alike

the terms *linux* and *modem* are used a lot in the first two docments. one can imagine that they are representive of a concept; we could call it *computers*
the terms *clutch*, *steering* and *petrol* are used a lot in the last three documents, perhaps they are representive of a concept; we could call it *automotive*
the term *the* is an interesting one; it used across all the documents and we can see that this is not really related to either topic, it's more an english construct

lsa will help up extract these features, *computers* and *automotive*
it won't though, alas, give us nice human readable names for them :)

next let's look at an example of svd

Aug 2009
me on twitter
me on google+

<< intro   index   example 2 >>

example 1: two clear features

here's another trivial example to make sure we know exactly what's going on

consider the set of, again, extremely contrived documents

```
d1: c a a b c b c
d2: a b c a b c c
d3: d e f f d
d4: f d e d f
```

which is represented as the 6x4 document term matrix.

	d1	d2	d3	d4
a	2	2	0	0
b	2	2	0	0
c	3	3	0	0
d	0	0	2	2
e	0	0	1	1
f	0	0	2	2

once more we can see two clear clusterings of the documents; d1 with d2 and d3 with d4

singular value decomposition

the singular value decomposition is a matrix decomposition algorithm (ie it breaks a matrix up into a series of products of matrices)

in particular the SVD decomposes a matrix A into the produce of three specially formed matrices (mysteriously named) U, S and V

each of these three matrices represents a different interpretation of the original data.

here is a decomposition of A performed using SVDLIBC

A = U x S x Vt

A

	d1	d2	d3	d4
a	2	2	0	0
b	2	2	0	0
c	3	3	0	0
d	0	0	2	2

U

	f1	f2	f3	f4
a	0.48	0	0	0
b	0.48	0	0	0
c	0.72	0	0	0
d	0	-0.66	0	0

S

	f1	f2	f3	f4
f1	5.83	0	0	0
f2	0	4.24	0	0
f3	0	0	0	0
f4	0	0	0	0

Vt

	d1	d2	d3	d4
f1	0.70	0.38	0	0
f2	0	0	-0.70	-0.70
f3	0	0	0	0
f4	0	0	0	0

A↓ U↓

e	0	0	1	1	e	0	-0.33	0	0
f	0	0	2	2	f	0	-0.66	0	0

S describes the relative strengths of the features
U describes the relationship between terms (rows) and features (columns)
Vt describes the relationshop between features (rows) and documents (columns)
even though the decomposition is expressed in terms of V transpose we'll usually talk about V so that the features are the columns in both U and V

interpretation of S

the matrix S is always a diagonal matrix with non-negative decending values, formally known as the singular values

each non zero value represents a feature and in this example we have two. (there can't be more features than there are documents) but we can have fewer features than docs
the magnitude of the values describes how much variance each feature describes in the data feature's strength
for this example we can see that each feature is roughly the same strength; there are roughly the same number of documents for each feature
(I'm not sure if the feature strength is related to the # of docs)

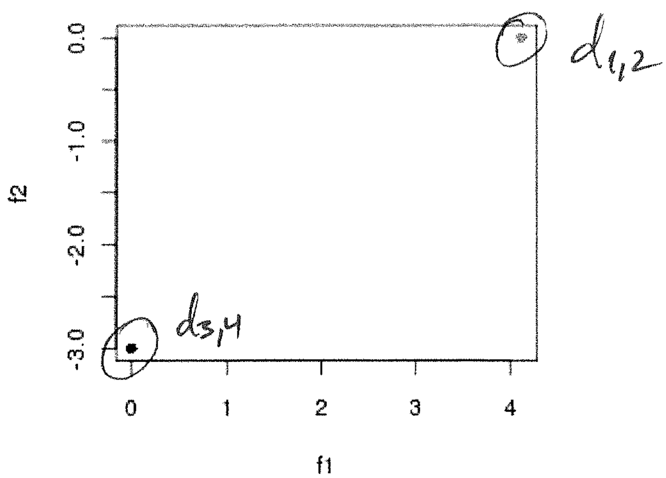
interpretation of VS

$= (SV^T)^T$

the matrix product VS describes the relation between documents (VS's rows) and the features (VS's columns) (SV^T)'s rows (SV^T)'s columns

plotting f1 vs f2 we can see the expected seperation of doc1 and doc2 from doc3 and doc4

	f1	f2	f3	f4
d1	4.123	0.000	0.000	0.000
d2	4.123	0.000	0.000	0.000
d3	0.000	-3.000	0.000	0.000
d4	0.000	-3.000	0.000	0.000



interpretation of US

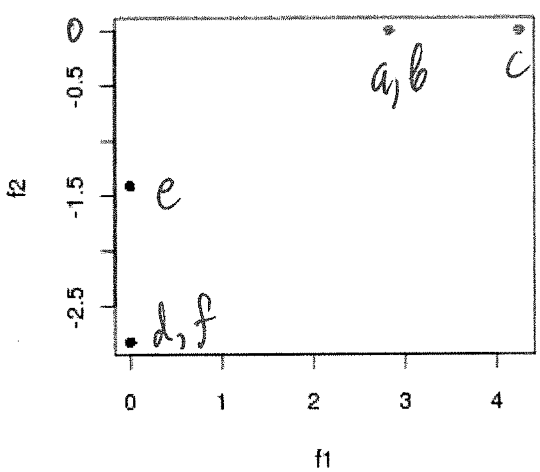
the matrix product US describes the relation between terms (US's rows) and the features (US's columns)

again we see what we expected; terms *a*, *b* & *c* (only present in the first two documents) are aligned with feature 1
whereas terms *d*, *e* & f (only present in the second two documents) are aligned with feature

2

we can also see that *c* has a stronger association with feature 1 than *a* or *b* does denoting that *c* occured more frequently
likewise *e* has a weaker association with feature 2 than *d* or *e* does denoting that *e* occured less frequently

	f1	f2	f3	f4
a	2.82	0	0	0
b	2.82	0	0	0
c	4.24	0	0	0
d	0	-2.82	0	0
e	0	-1.41	0	0
f	0	-2.82	0	0



let's now look at slighty more complex example

<< intro index example 2 >>

Aug 2009
me on twitter
me on google+

example 2: two less clear features

here's a slightly more complex example to work with

consider the documents

```
d1: c a a b c b c
d2: a b c a b c c
d3: d e c f c f d c
d4: c c f d e d f
```

which are represented as the 6x4 document term matrix.

	d1	d2	d3	d4
a	2	2	0	0
b	2	2	0	0
c	3	2	3	2
d	0	0	2	2
e	0	0	1	1
f	0	0	2	2

once more we can see a partitioning of the documents; d1 with d2 and d3 with d4
like our original example it's not as clear cut since *c* is present in d1 and d2 as much as it is in d3 and d4.

singular value decomposition

here is a decomposition of A performed using SVDLIBC

A **=** **U** **x** **S** **x**

	d1	d2	d3	d4
t1	2	2	0	0
t2	2	2	0	0
t3	3	2	3	2
t4	0	0	2	2
t5	0	0	1	1
t6	0	0	2	2

	f1	f2	f3	f4
t1	0.292	-0.503	0.402	0.000
t2	0.292	-0.503	0.402	0.000
t3	0.778	-0.048	-0.626	0.000
t4	0.316	0.467	0.356	0.000
t5	0.158	0.233	0.178	0.000
t6	0.316	0.467	0.356	0.000

	f1	f2	f3	f4
f1	6.52	0	0	0
f2	0	4.11	0	0
f3	0	0	0.63	0
f4	0	0	0	0

Vt

	d1	d2	d3	d4
f1	0.536	0.417	0.575	0.456

f2	-0.524	-0.513	0.475	0.487
f3	-0.433	0.560	-0.440	0.553
f4	0.398	16.962	42.639	0.000

recall:
S describes the relative strengths of the features
U describes the relationship between terms (rows) and features (columns)
Vt describes the relationshop between features (rows) and documents (columns)
even though the decomposition is expressed in terms of V transpose we'll usually talk about V so that the features are the columns in both U and V

interpretation of S

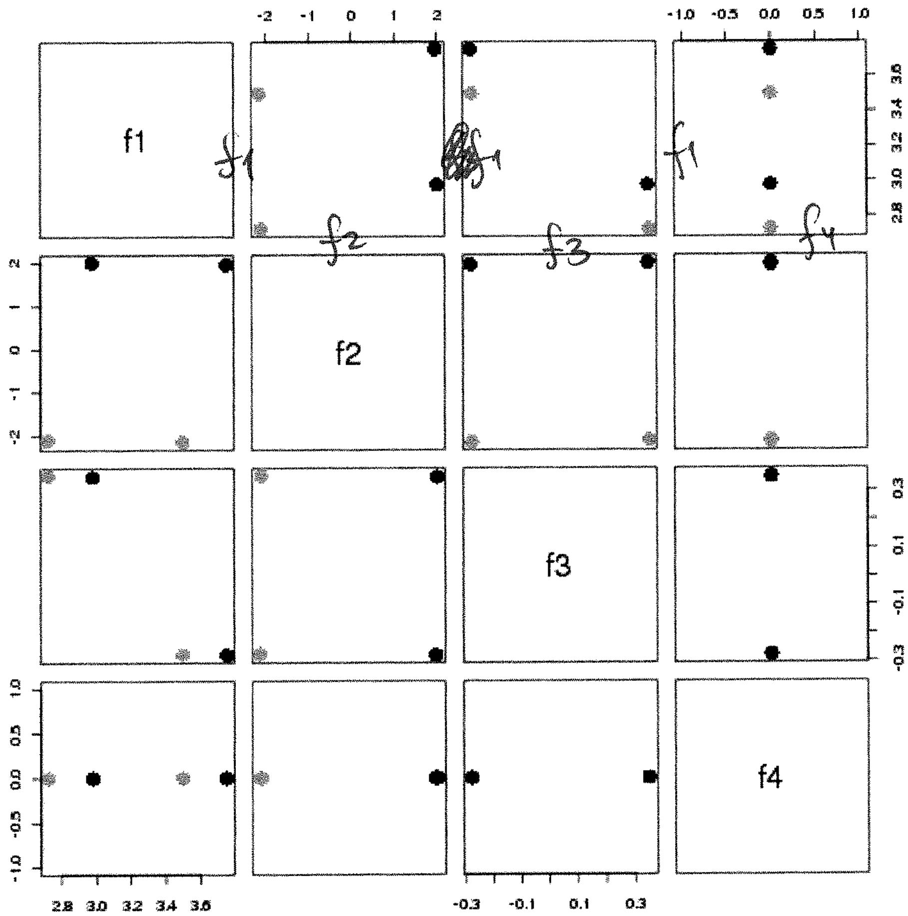
this time we have 3 singular values; 2 dominant ones (f1 and f2) and 1 lesser one (f3) so again the variance of this data is primarily described by 2 features

interpretation of VS

recall the matrix product VS describes the relation between documents (VS's rows) and the features (VS's columns) (SV^T)'s rows (SV^T)'s columns

it's not as straight forward as our the last example
this time the dominant feature f1 describes not a type of document but the use of the common term c // ← not obvious; take on faith
it's f2 that gives a clear seperation of d1 and d2 from d3 and d4
the scatterplot matrix below seems to suggest in this case that f2 alone is the best distinguisher of the two types of documents.

	f1	f2	f3	f4
d1	3.502	-2.159	-0.273	0.000
d2	2.724	-2.111	0.353	0.000
d3	3.755	1.956	-0.278	0.000
d4	2.977	2.005	0.349	0.000



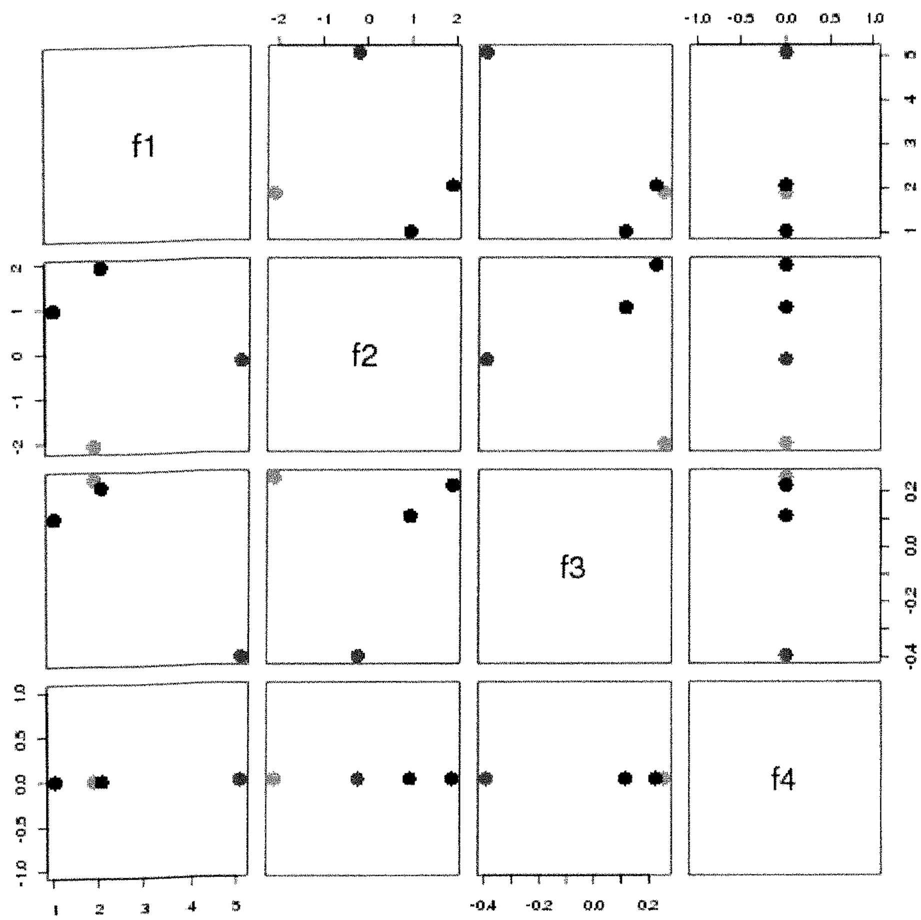
interpretation of US

recall the matrix product US describes the relation between terms (US's rows) and the features (US's columns)

as above we see that the strongest feature f1 is primarily related to the term *c*
f2 gives a reasonable separation for the 3 types of terms in the corpus;

1. those that are strongest in d1 and d2 (green),
2. those that are shared (red)
3. those that are strongest in d3 and d4 (blue)

	f1	f2	f3	f4
a	1.907	-2.074	0.253	0.000
b	1.907	-2.074	0.253	0.000
c	5.080	-0.200	-0.395	0.000
d	2.062	1.923	0.225	0.000
e	1.031	0.962	0.112	0.000
f	2.062	1.923	0.225	0.000



where as the first example was a simple case of 1 feature = 1 type of document, it's more complex this time
the first feature instead describes the use of a very common term which apparently is quite common.
the highest features often relate to language semantics, with later features describing corpus structure

let's move onto an even more complex example

<< example 1 index example 3 >>

Aug 2009
me on twitter
me on google+

example 3: two less clear features (revisited)

let's move onto an example, similiar to the last, but with some more 'noise'

again let's work with a contrived corpus

```
d1: a a a a a a  b b b b b b b b  c c c c c c  e e  f f
d2: a a a a a a a  b b b b b b  c c c c c c c c c  d
d3: a  c c c c c c c c  d d d d d d d d  e e e e e e e e e  f f f f f f f
d4: b  c c c c c  d d d d d d d d  e e e e e e e  f f f f f f f
```

which is represented as the 6x4 document term matrix

	d1	d2	d3	d4
a	6	7	1	0
b	8	6	0	1
c	6	9	8	5
d	0	1	8	8
e	2	0	9	7
f	2	0	7	7

singular value decomposition

here is a decomposition of A performed again using SVDLIBC

A = U x S x

A

	d1	d2	d3	d4
a	6	7	1	0
b	8	6	0	1
c	6	9	8	5
d	0	1	8	8
e	2	0	9	7
f	2	0	7	7

U

	f1	f2	f3	f4
a	0.24	-0.51	0.08	0.06
b	0.25	-0.54	-0.64	-0.23
c	0.58	-0.28	0.57	0.13
d	0.42	0.37	0.16	-0.68
e	0.44	0.34	-0.24	0.66
f	0.39	0.29	-0.40	-0.09

S

	f1	f2	f3	f4
f1	23.1	0	0	0
f2	0	14.3	0	0
f3	0	0	3.5	0
f4	0	0	0	1.5

Vt

	d1	d2	d3	d4
f1	0.37	0.38	0.65	0.53
f2	-0.55	-0.63	0.37	0.38
f3	-0.69	0.59	0.27	-0.21

f4	0.26	-0.29	0.59	-0.69

recall:
S describes the relative strengths of the features
U describes the relationship between terms (rows) and features (columns)
Vt describes the relationshop between features (rows) and documents (columns)

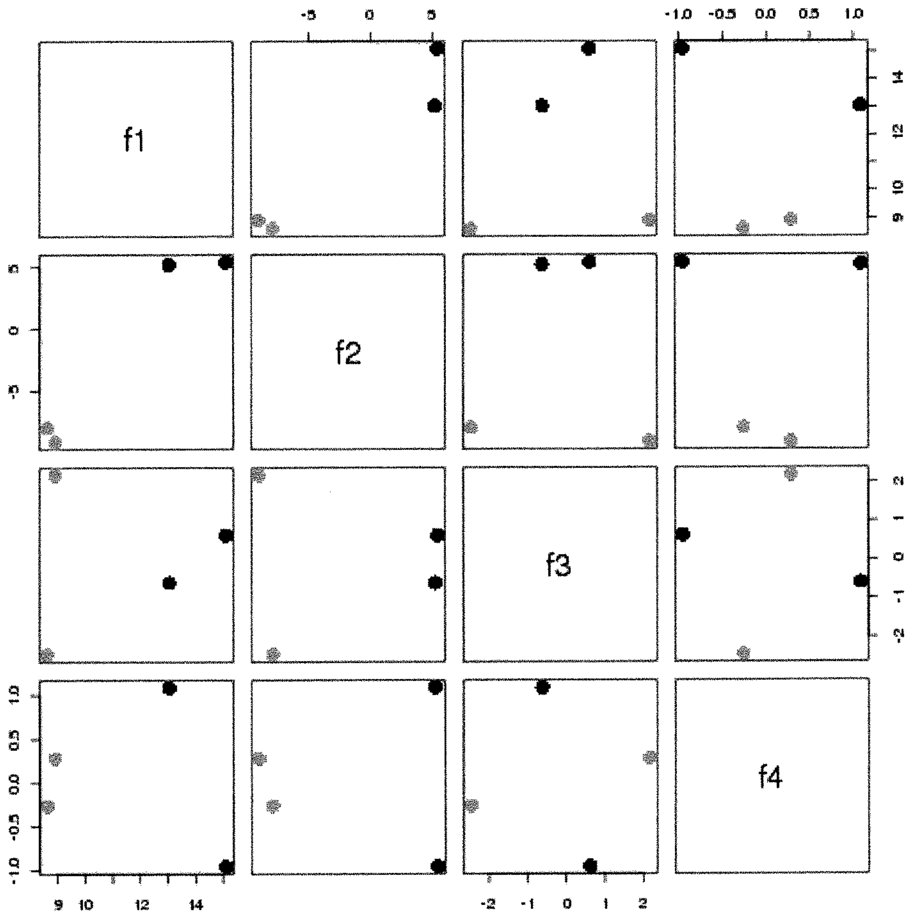
interpretation of S

similiarly to our last example we've got two dominant features but the additional non zero term frequencies have meant we've got variance for all possible 4 features.
(this is representive of the general non contrived case where if we were dealing with a large number of documents we'd only be interested in the first dominant features)
like last time since the first two values are much higher than the second two we can derive that there are two main dominant features in the corpus.

interpretation of VS

the matrix product VS describes the relation between documents (VS's rows) and the features (VS's columns)

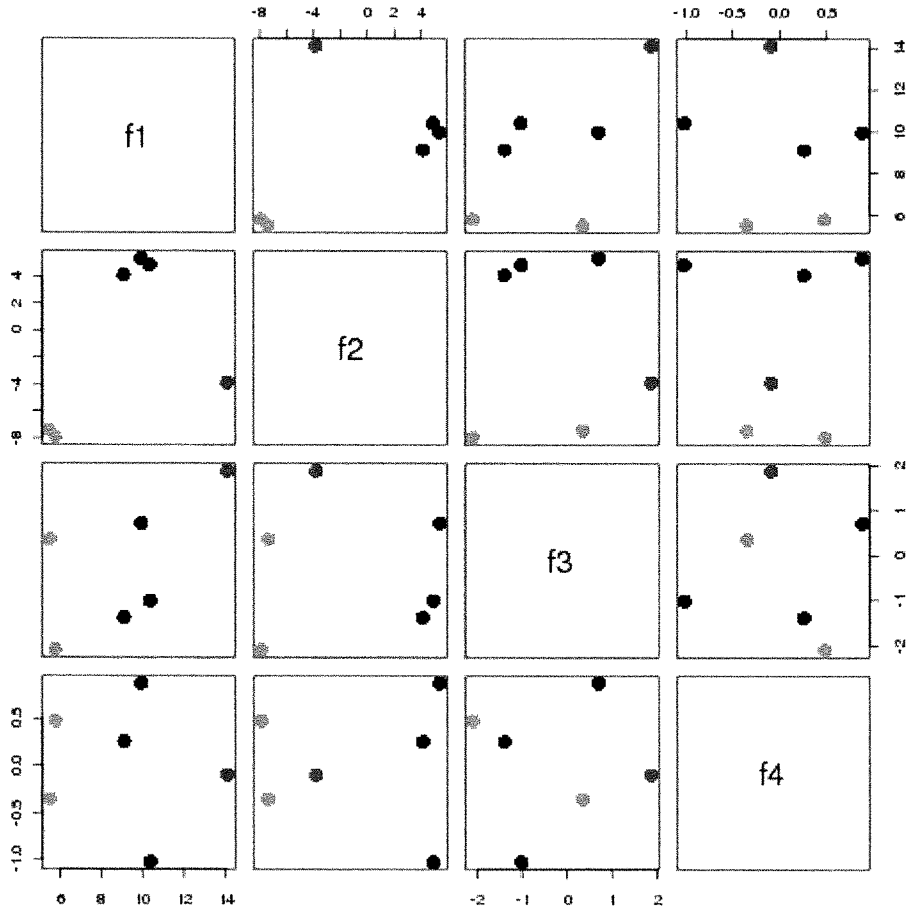
	f1	f2	f3	f4
d1	8.624	-7.973	-2.447	-0.259
d2	8.928	-9.081	2.177	0.283
d3	15.116	5.402	0.627	-0.956
d4	13.044	5.227	-0.599	1.086



interpretation of US

the matrix product US describes the relation between terms (US's rows) and the features (US's columns)
there is a bit more jitter in the points but similiar analysis as last time holds

	f1	f2	f3	f4
a	5.502	-7.449	0.349	-0.352
b	5.768	-7.943	-2.100	0.477
c	14.091	-3.864	1.869	-0.095
d	9.962	5.337	0.709	0.880
e	10.404	4.867	-1.016	-1.019
f	9.118	4.107	-1.386	0.259



ok then, enough of this contrived stuff, let's have a look at an example with real data

<< example 2   index   real data example >>

real data example

the corpus

let's try the decomposition on some real data and see what patterns we find

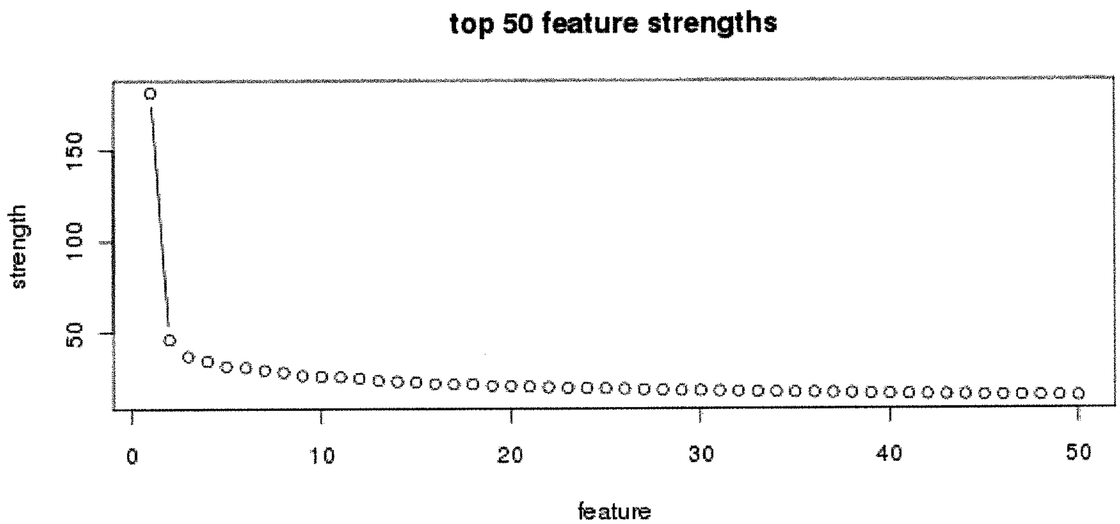
we'll use a simple dataset of 100 articles taken from each of 3 quite different rss feeds;

- autoblog (a automotive discussion blog)
- perez hilton (a hollywood gossip blog)
- the register (a tech review blog)

we should be able to find enough variance in features to be able to classify a new article as coming from one of these three.

feature strengths

first let's look at the feature strength for the first 50 features



it seems pretty clear that the first feature is the major one

the first feature

terms related to the first feature

of the 5700 terms present in the corpus which terms are strongest for the first feature?

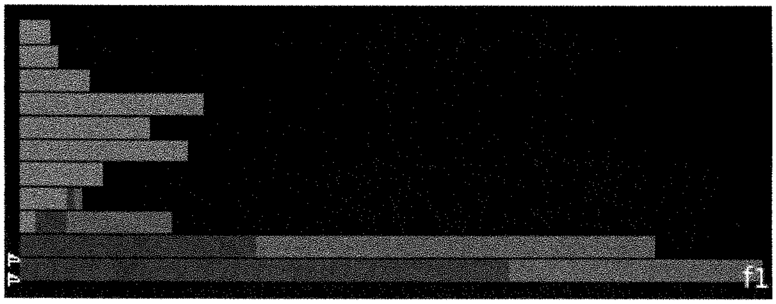
rank	1	2	3	4	5	6	7	8	9	10
term	the	of	to	and	in	for	that	is	with	it
strength	138	46	45	43	32	25	25	22	16	16

at the tail end there are the hapax legomenon with near zero scores including terms like... un, sydney, soa, jailed, worker, diplomat

to me this indicates a feature pretty strongly associated with common english constructs (apparently this is quite common in LSA)
if nothing else then SVD is an extremely expensive way to do language determination :)

documents related to the first feature

given we've seen that the features describe english terms we should expect it to be pretty arbitrary which documents are most strongly associated with this feature. let's see.

<table>
<tr><td colspan="3">feature 1 article strengths</td></tr>
<tr><td colspan="3">(articles near top most strongly associated)</td></tr>
<tr><td colspan="3">
</td></tr>
<tr><td>autoblog</td><td>the register</td><td>perez hilton</td></tr>
</table>

we can see the the first feature is most strongly, and exclusively, associated with the articles from autoblog
articles for the register and perez hilton> are less associated (the bottom bars of the histogram)

if this feature corresponds to english constructs why is it so strongly associated only with autoblog?
seems that the autoblog articles on average are much longer than the other two feeds.

feed	total terms in corpus
autoblog	19347
perez	4392
the register	2658

does this imply we'll have to normalise the data in some way first? we'll come back this ...

the second feature

terms related to the second feature

the terms most strongly associated on the +ve side with the second feature are quite similiar to the common language terms of the first feature

rank	1	2	3	4	5	6	7	8	9	10
feature2	and	of	in	that	for	is	the	on	gallery	you
strength	0.45	0.43	0.27	0.20	0.17	0.16	0.13	0.12	0.12	0.11

but the terms most strongly associated on the -ve side *do* show something...

rank	5718	5719	5720	5721	5722	5723	5724	5725	5726	5727

feature2	opportunity	not	had	weekend	show	very	new	this	we	cher
strength	-0.99	-1.00	-1.00	-1.00	-1.00	-1.01	-1.02	-1.02	-1.05	-45.98

cher? with an overwhelming strength of -45?!?!

documents related to the second (aka cher) feature

in the same way there is a single term dominanting the second feature
there is a single document, from perezhilton, that dominates the second feature

```
Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher!
Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher!
Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher!
Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher! Cher!
Cher! This weekend we had the very special opportunity to not only
see Cher's new show ...
```

so in fact this second feature is not related to a *type* of article but just this particular article
this makes me think even more that we need some normalisation, but let's continue for a few more features

features three and four

features 3 and 4 are similiar to feature 2 in that they're associated again to a single article, this time one from autoblog.

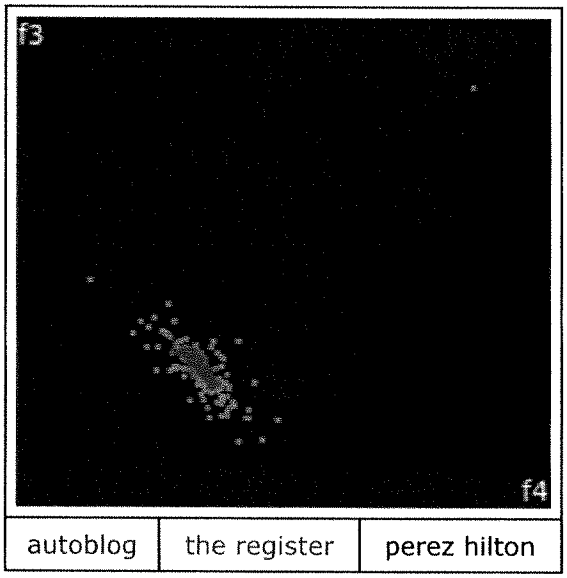
terms related to the third and fourth feature

rank	1	2	3	4	5	6	7	8	9	10
feature3	to	and	in	sales	20	comparechart	34	chrysler	14	24
strength	14.3	14.1	6.75	6.0	4.8	4.7	4.0	3.5	3.4	3.3
feature4	the	sales	20	comparechart	34	25	14	24	in	audi
strength	8.5	5.8	4.5	4.5	3.9	3.8	3.3	3.2	3.0	3.0

document related to the third and fourth feature

the autoblog article relating to these two features is by far the longest (in terms of raw chars) since
it includes a nested table that wasn't parsed out very well by my original slurping script

feature 3 vs feature 4 scatterplot



Filed under: By the Numbers Check it out. We've completely revamped By the Numbers to convey more sales information than before in a much easier to digest way. Now we'll be reporting both the change in monthly sales volume for each brand and automaker as well as the change in their Daily Sales Rate or average number of vehicles sold per day. On to the armchair analysis... Poor sales continued through the month of August as only a handful of brands are able to brag about increased sales. Nissan North America bucked the trend entirely reporting a 13.6% gain for the combined brands of Nissan and Infiniti with each marque reporting its own individual increases. Credit goes to VW (2.9%), as well, which posted a solid number, and the BMW Group (1.0%), which barely earned a positive increase in sales thanks to a strong 34.1% increase in MINI sales. While GM (-20.4%), FoMoCo (-25.6%) and Chrysler LLC (-34.5%) sales were all down in a big way, Toyota MoCo and Honda America were also not immune falling 9.4% and 7.3%, respectively. In this environment, brands should consider a single-digit drop a small victory considering the majority of brands that fell by 10% or more. #comparechart { border: 2px solid #333; border-collapse: collapse; } #comparechart td { padding: 3px; border: 1px solid #ccc; vertical-align: top; margin: 0; line-height: 1.3em; font-size: 80%} #comparechart th { font-size: 80%; font-weight: bold; text-align: left; padding: 4px; background: #eee; } #comparechart th.mainth { font-size: 75%; border-bottom: 1px solid #333; } #comparechart td.red { background-color: #f08c85; } #comparechart td.green { background-color: #b3e2c4; } #comparechart td.yellow { background-color: #ffffcc;} BY THE NUMBERS - August 2008 Brand Vol. Total Vol. 8/08 Total Vol. 8/07 DSR Daily avg 8/08 Daily avg 8/07 Acura -8.2% 15,089 16,436 -8.2% 559 609 Audi -15.9% 6,406 7,620 -15.9% 237 282 BMW -4.1% 25,462 26,562 -4.1% 943 984 Buick -7.7% 17,833 19,324 -7.7% 660 716 Cadillac -20.9% 15,405 19,481 -20.9% 571 722 Chevrolet -19.2% 185,080 229,012 -19.2% 6,855 8,482 Chrysler -44.2% 24,337 43,650 -44.2% 901 1,617 Dodge -24.6% 62,422 82,841 -24.6% 2,312 3,068 Ford -26.2% 133,088 180,282 -26.1% 4,929 6,677 GMC -17.6% 42,194 51,222 -17.6% 1,563 1,897 Honda -7.2% 131,766 141,906 -7.2% 4,880 5,256 HUMMER -62% 2,160 5,677 -62% 80 210 Hyundai -8.8% 41,130 45,087 -8.8% 1,523 1,670 Infiniti 8.0% 11,076 10,252 8.0% 410 378 Jeep -43.7% 23,476 41,712 -43.7% 869 1,545 Kia -6.7% 25,065 26,874 -6.7% 928 995 Lexus -9.1% 29,281 32,199 -9.1% 1,084 1,193 Lincoln -8.5% 9,540 10,423 -8.5% 353 386 Mazda -4.4% 23,680 24,762 -4.4% 877 917 Mercedes-Benz -11.8% 18,507 20,980 -11.8% 685 777 Mercury -31.7% 8,393 12,296 -31.7% 311 455 MINI 34.1% 5,469 4,077 34.1% 203 151 Mitsubishi -29.3% 9,200 13,020 -29.3% 341 482 Nissan 14.2% 97,417 85,275 14.2% 3,608 3,158 Pontiac -38.3% 24,257 39,324 -38.3% 898 1,456 Porsche -44.9% 1,404

2,548 -44.9% 52 94 Saab -50.1% 1,503 3,011 -50.1% 56 112 Saturn -3.5% 20,385 21,117 -3.5% 755 782 Subaru 14.2% 18,932 16,573 14.2% 701 614 Suzuki -31.7% 6,083 8,916 -31.7% 225 330 Toyota -9.4% 182,252 201,272 -9.4% 6,750 7,455 Volkswagen 2.9% 22,292 21,655 2.9% 826 802 Volvo -48.8% 4,669 9,119 -48.8% 173 338 COMPANIES BMW Group 1% 30,931 30,639 1% 1,146 1,135 Chrysler LLC -34.5% 110,235 168,203 -34.5% 4,083 6,230 FoMoCo -25.6% 151,021 203,001 -25.6% 5,593 7,519 General Motors -20.4% 308,817 388,168 -20.4% 11,438 14,377 Honda America -7.3% 146,855 158,342 -7.3% 5,439 5,864 Nissan NA 13.6% 108,493 95,527 13.6% 4,018 3,538 Toyota Mo Co -9.4% 211,533 233,471 -9.4% 7,835 8,647 August 2008 had 27 selling days versus 27 selling days for August 2007 UPDATE: Audi added and Subaru's sales figures corrected. ? Permalink | Email this | Comments

ouch. even more ammo for some pre normalisation step

features from five onwards

terms related to these features

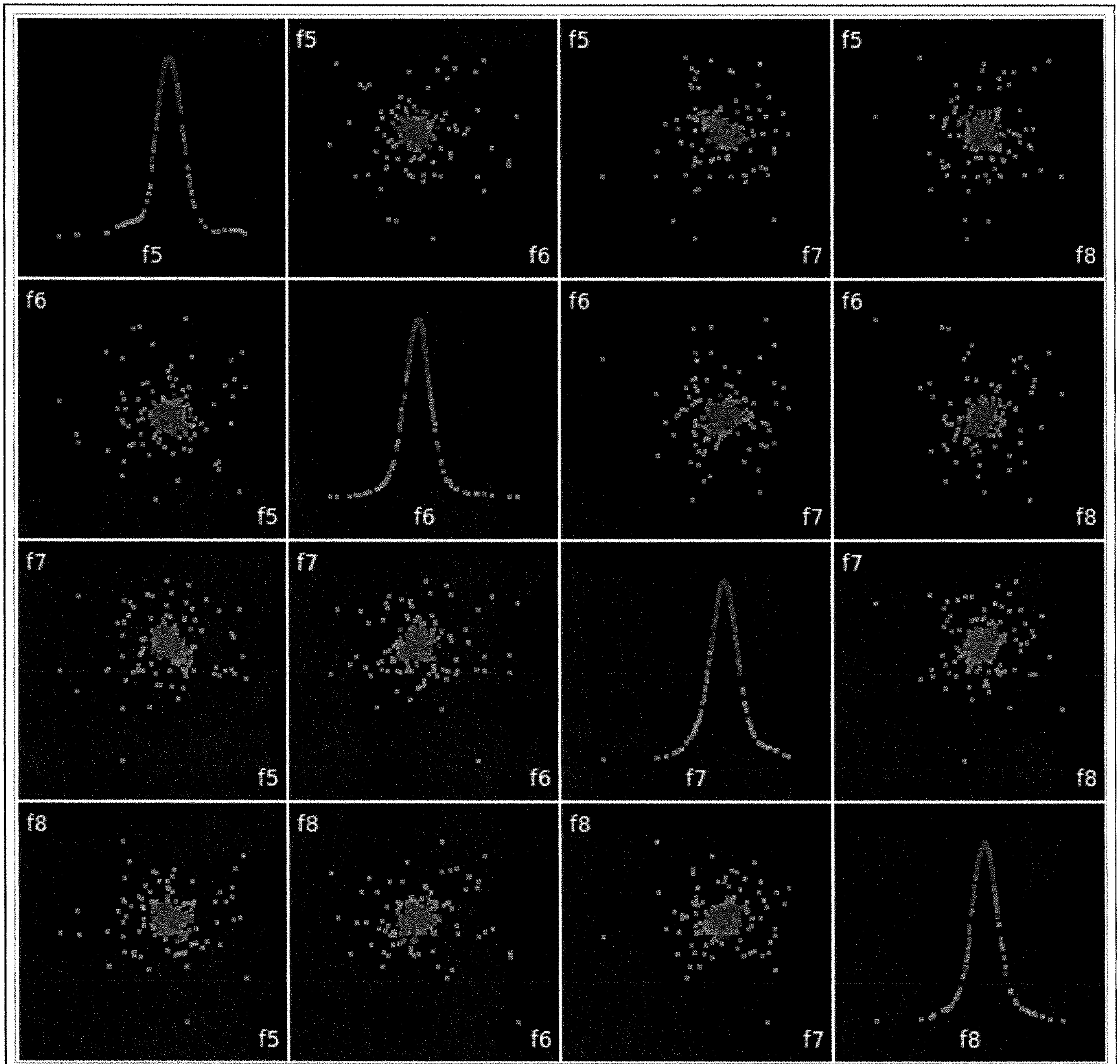
nothing really sticks out for these features...

documents related to these features

the 10 most +ve and -ve documents for features 5 onwards are from autoblog with those articles dominating the edges of the feature space
articles for the register and perez hilton cluster around 0.
i suspect this is again an artifact of the longer autoblog articles.

we can see that in the following scatterplot matrix that autoblog entries encircle the others.
i'm a sure a pretty vanilla svm would pick this up boundary
if it's just document length that is the reason for this spread a much simpler classifier would be to just check the article length.

feature 5 to feature 4 scatterplot matrix



autoblog	the register	perez hilton

so it really looks like we need to normalise the input in some way.
let's try the most vanilla we can, just normalising on the doc length

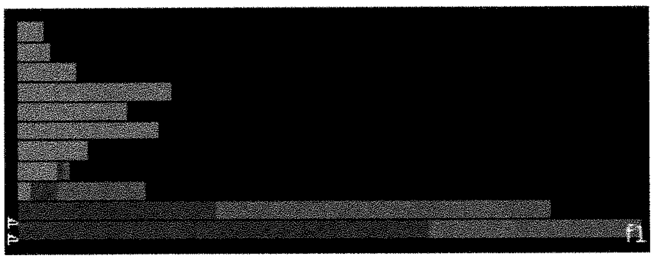
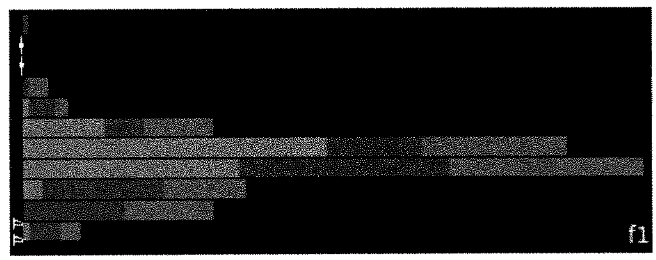
<< example 3 index real data example (with normalisation) >>

<< real data example index conclusions >>

real data example (normalised doc lengths)

this time let's so the same but include a *really* simple normalisation; divide each term's weight based on the document length.

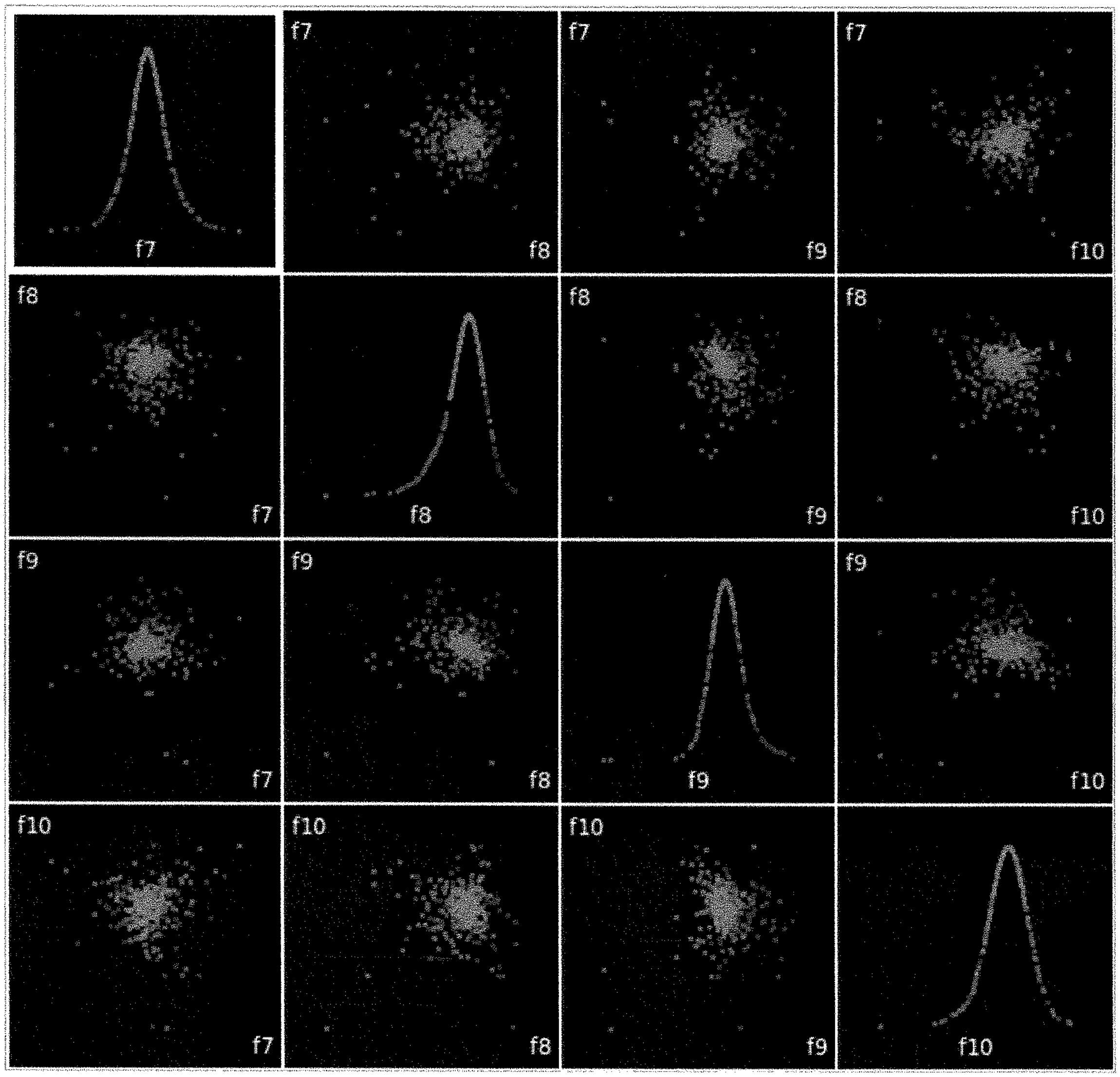
when we do this we get an immediate improvement on the spread for the first feature.
like last time it consists of english construct words but this time isn't dominated by autoblog articles

<table>
<tr><td colspan="3">feature 1 article strengths</td></tr>
<tr><td colspan="3">(articles near top most strongly associated)</td></tr>
<tr><td colspan="2">without normalisation</td><td>with normalisation</td></tr>
<tr><td colspan="2">
</td><td>
</td></tr>
<tr><td>autoblog</td><td>the register</td><td>perez hilton</td></tr>
</table>

and like last time the following few features (f2 to f6) are related to single documents which have some fundamental difference in them to the entire corpus

features 7 through 10 show an interesting seperation of the documents
consider especially f8 vs f9

feature 7 to feature 9 scatterplot matrix



autoblog	the register	perez hilton

here's an undirected 2d tour of the feature space for features 7 through 10, seems to be quite a bit of seperation.

feature 7 to feature 9 scatterplot matrix		
autoblog	the register	perez hilton

so finally, some conclusions

<< real data example index conclusions >>