Lab 4: Dimensionality Reduction

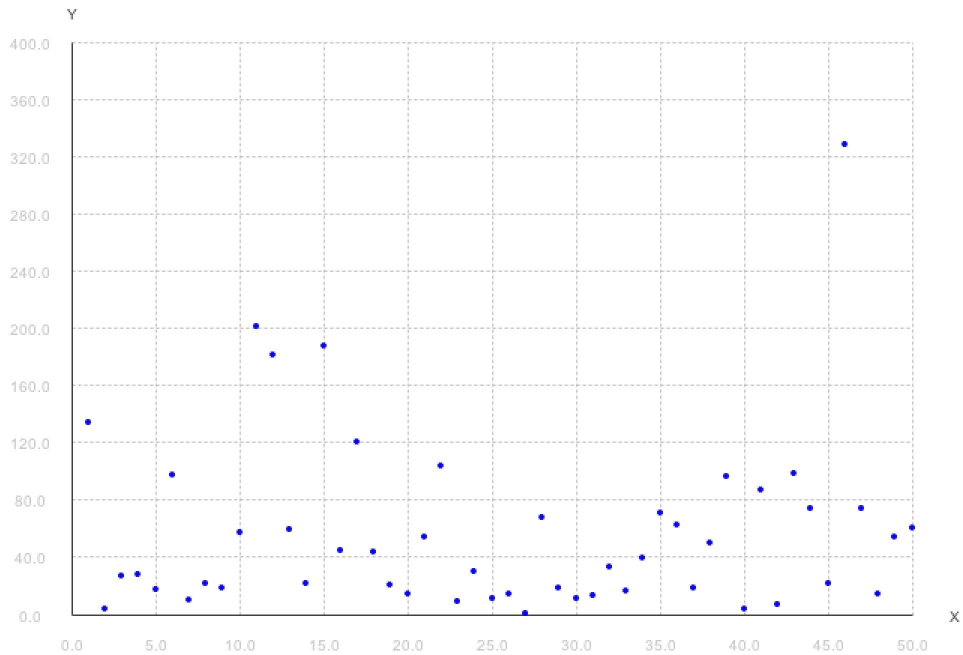
Ruofan Zhou

March 24, 2014

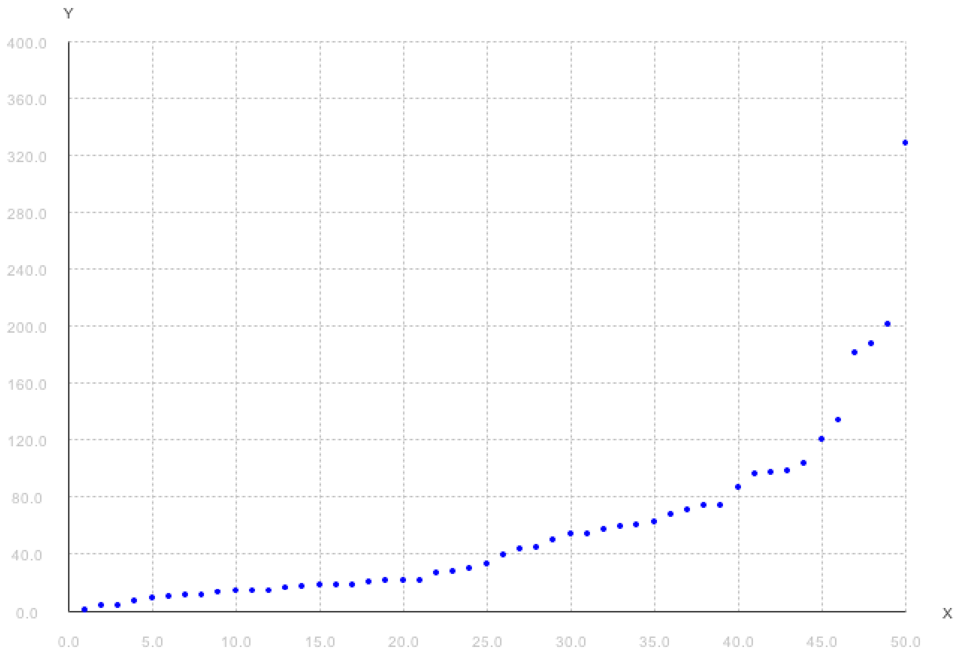
1 Faces

1.1 Dimensionality of the Dataset

1. There are *5000* faces and *50* measurements.
2. See the *variance-dimension* plot as below:



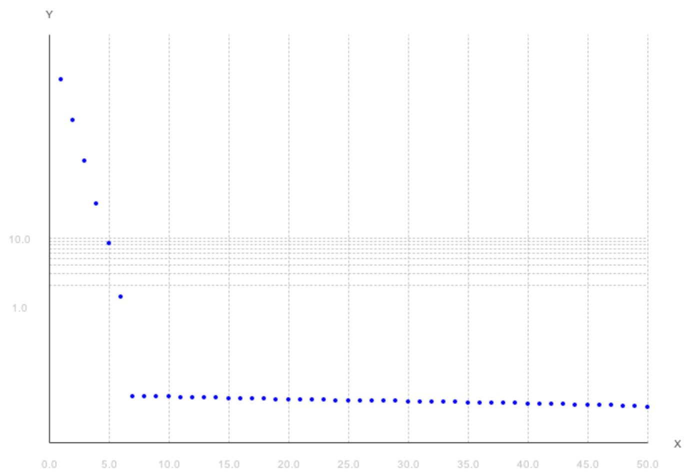
3. I perfer to describe the face with *7* variable parts: *hat color, hair length, hair color, eye size, face color, mouth shape, eye color.*
4. I sorted the variances and get the plot:



but still can't really find how many dimensions it can be precised to.
5. It's really hard and I give it up.

1.2 Principal Component Analysis

6. Done.
7. See the plot of the variances as below. It's obvious that *7* components are stand out, and it's equal to the answer of quesion 2.



8. & 9. Done.

10. With very happy(mouth up) face[dim 1], whose eyes are slightly larger than normal[dim 2] and green[dim 3]. His hair is relatively[dim 4] and is in bright yellow color[dim 5]. Normal face color(not too dark or too bright)[dim 6], and pobably with a brown hat[dim 7].

2 Smartvote

See file *CoActorPrediction.java* to view my implementation of the classes.
And I computer *MyOwnScorning* with:

$$\frac{N(u,v)^2 \times \sqrt{|N(u) - N(v)|}}{N(u) + N(v)}$$

In which N(i) stands for neibour number of i, and N(i, j) stands for common neibour number of i and j.
And Here's accuracy of the 3 link prediction strategies:

Name	Accuracy	Improvement
PreferentialAttachment	2.096%	3.19x
CommonNeighbors	5.114%	7.79x
MyOwnScoring	5.388%	8.21x

3 Random Walks

1. Setting N = 10000 and node u as initial seed, the average of a Facebook user is *48.6725*, which still has a difference with the reference answer. The problem I think, is the *algorithm itself*, and *maybe node u is in a small component of the network*, and maybe *the dataset itself changes.*
2. I changed walk strategy in my algorithm: every time I walk across 3 people instead of one(i.e. I walk to the node's neighbor's neighbor's neighbor instead of its neighbor), and there's also a repeat judgment in my code. After changing, the estimated age becomes *50.675*, which is more close to the reference answer.

3.1 Bizarre Social Networks

1. Setting N = 10000 and using different beginning node, we get average age as below:

we can see much difference between each other. And we can also find that with younger beginning node we get smaller average age and with older beginning node we get larger average age. Maybe it's the *Herding* property. The friends are seems so familiar.

2. The same strategy as above, now get the answers:

   The answeres now become more similar :)

Finally, I use my strategy(this time, walk across 6 people every time) on the *Directions*, get answers:

So, my estimate of the average age of Directions's users is *49.5*.

Beginning Node	Average Age	Age of Beginning Node
u	50.1445	13
v	55.5635	39
w	64.1295	89

Beginning Node	Average Age	Age of Beginning Node
u	58.721	13
v	58.2856	39
w	57.4979	89

Beginning Node	Average Age	Age of Beginning Node
u	49.5636	23
v	49.4373	22
w	49.4054	39